AA-DK72A-TE

WPS-PLUS/ ALL-IN-1

List Processing

digital™

AA-DK72A-TE
First Printing, February 1985

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

Copyright© 1985 by Digital Equipment Corporation, Maynard, Mass.

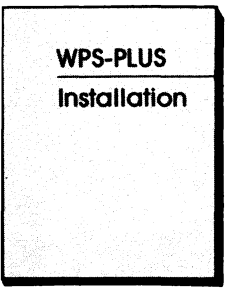
The postage-prepaid Reader's Comments Form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

digital™
DECspell
DECmate
DECsystem-10
DECSYSTEM-20
DECUS
DECwriter
DIBOL
MASSBUS
PDP
P/OS
Professional
Rainbow
RSTS
ALL-IN-1
RSX
UNIBUS
VAX
VMS
VT
Work Processor
WPS-PLUS

WPS-PLUS/ALL-IN-1

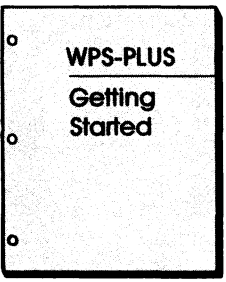
Documentation Chart



This guide is for the installer of the WPS-PLUS software.

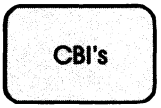
The guide tells you how to prepare for the installation and how to install WPS-PLUS on your operating system. It includes instructions for a test to ensure that WPS-PLUS is properly installed.

Before you can use WPS-PLUS or the CBIs, WPS-PLUS must be installed.



This guide is for new WPS-PLUS users.

The guide uses lessons to help you practice some basics, such as creating, editing, and printing a document. It shows you how to insert and remove text, move the cursor, and use key editing functions.

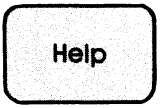


Computer-Based Instructions (CBIs) are for new users and users who want a refresher course on WPS-PLUS basics.

CBI courses give you a system overview and let you try using WPS-PLUS. CBI training is available from the WPS-PLUS Main Menu.

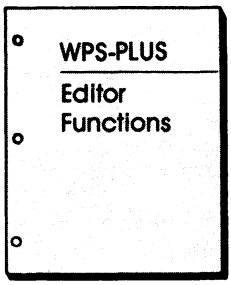
Screen messages are for all users.

WPS-PLUS displays messages on your screen to let you know what options are available, what data is needed, and, when possible, to help you correct a data entry error.



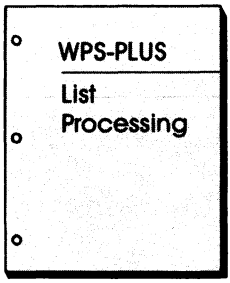
Help is for all WPS-PLUS users.

When you ask for help, a message helps you complete your task, such as filling in a form field, or entering a menu option. Help is always available.



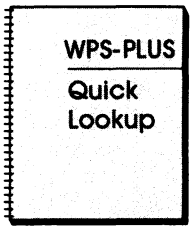
This guide is for WPS-PLUS users who understand the basics as explained in Getting Started and the CBIs.

The guide explains how to use all editor functions.



This guide is for WPS-PLUS users who understand the basics as explained in Getting Started and the CBIs.

The guide shows how to create and combine lists of information with forms to produce customized documents. It shows you how to use the List Processing Math and Sort features.



This guide is for WPS-PLUS users who want to quickly review the steps used in WPS-PLUS operations and functions.

Contents

About This Guide

Purpose of This Guide xi
Audience xi
Prerequisites xii
Associated Documents xii
Conventions xiii

1 List and Sort Processing

What is List Processing? 1-1
  How List Processing Works 1-2
  How to Create a List Document 1-4
  How to Create a Form Document 1-5
  How to Create a List Specification Document 1-6
  How to Run List Processing 1-6
  How to Get Help 1-7
What is List Processing Math? 1-7
  How List Processing Math Works 1-8
  How to Create a FORMULA Control Block 1-8
What is Sort Processing? 1-10
  How Sort Processing Works 1-11
  How to Create a Sort Specification Document 1-12
  How to Run Sort Processing 1-12

2 Preparing Your Lists

List Documents 2-2
Records 2-2
Fields 2-2
  Typing Field Names 2-2
  Typing Field Values 2-3
List Document Guidelines 2-3
  List Size 2-4
  Order of Records 2-4
  Order of Fields 2-4
  Using Numeric Fields 2-4
  Entering Dates as Field Values 2-4
  Missing Fields and Field Values 2-5
  Inserting Comments 2-5
  Using Abbreviation and Library Documents 2-6

3 Formatting Your Output

Form Documents 3-1
Types of Form Documents 3-2
  Continuous-Form Letterhead 3-3
  Continuous-Form Envelopes 3-5
  Continuous-Form Labels 3-7
  Tables 3-8
Form Document Commands 3-9
Leaders and Trailers 3-9
Spacing Leaders and Trailers 3-12
Spacing Table Entries 3-13
  Single and Double Spacing 3-13
  Grouping Entries 3-15
Columns 3-16
Skipping Records 3-18
Copying Lists 3-18
Changing List Document Field Names 3-19

4 Selecting the Records to Process

List Specification Documents 4-1
Restricting the Records You Process 4-2
  Testing For Any of Several Values 4-3
  Testing For More Than One Value 4-3
  Rejecting Records 4-4
  Using More Than One IF-Statement 4-5
  Testing for Empty Fields 4-5
  Testing with Wild Cards 4-6
  Testing Field Values in Lists 4-8
  Comparing Numeric Fields 4-8
  Comparing Character-For-Character 4-9
  Comparing Digits Only 4-9
  Testing Against Numeric Ranges 4-10
  Testing Dates 4-11
  Testing Dates Against a Range 4-12
  List Specification Guidelines 4-12
Testing a List Specification Document 4-13
Words and Phrases in a List Specification:
  Summary 4-14

5 List Processing Math

List Processing Math and Editor Math 5-1
List Processing Math Commands 5-2
Starting List Processing Math 5-3
Stopping List Processing Math 5-4
FORMULA Command Guidelines 5-5
Allowable Characters 5-6
Constants 5-7
Processing List Processing Math 5-7
List Processing Math: Complete Example 5-7
Totals and Subtotals 5-9
Averages 5-12
Default Format 5-14
Customized Format 5-15
  Decimal Fractions 5-15
  Commas 5-16

Dollar Signs 5-16
Trailing Minus Sign 5-16
Parentheses 5-17
Asterisks 5-17
Truncation and Rounding 5-18
Records Selected for Processing 5-19
List Processing Math Error Messages 5-19
FORMULA Command Checklist 5-20
Format Checklist 5-21
Input Field Checklist 5-21

6 Sorting a List Document

Sort Specification Documents 6-1
Sorting on More Than One Field 6-2
Examples of Sort Output Documents 6-4
Sort Alphabet 6-9
Sorting Missing Fields or Field Values 6-10
Words and Phrases in a Sort Specification:
Summary 6-10

7 Running List and Sort Processing

List Processing Menu 7-1
List Processing 7-3
Completing the Document Selection Forms 7-3
How to Start List Processing 7-7
Start and Finish Settings 7-8
Error Messages 7-9
Printing an Output Document 7-9
Sort Processing 7-10
Completing the Document Selection Forms 7-10
How to Start Sort Processing 7-11

Glossary

Index

Figures

1-1 How List Processing Works 1-3
1-2 How List Processing Math Works 1-9
1-3 How Sort Processing Works 1-11
4-1 Words and Phrases in a List Specification 4-14
5-1 List Processing Math Commands 5-8
5-2 Subtotals 5-10
6-1 List Sorted on <STATE>, <CITY>, and <STREET> Ascending Order 6-3
6-2 List Sorted on <STATE> and <LAST> in Ascending Order 6-5
6-3 List Sorted on <ITEM#> in Ascending Order 6-6
6-4 List Sorted on <PAYMT> Descending and <LAST> Ascending 6-7
6-5 List Sorted on <:UNIT PRICE> in Descending Order 6-8
6-6 Words and Phrases in a Sort Specification 6-10
7-1 List Processing Menu 7-2
7-2 Selecting List Document Form 7-4
7-3 Selecting Output Document Form 7-6
7-4 Creating New Output Document Form 7-6
7-5 List Processing Documents Selected Form 7-7

Tables

3-1 Form Document Commands 3-9
3-2 Single and Double Spacing with <!S> and <!E> 3-14
4-1 Words and Phrases in a List Specification 4-15
5-1 Editor Math and List Processing Math 5-2
5-2 List Processing Math Commands 5-3
5-3 Allowable Characters 5-6
6-1 Sort Alphabet 6-9
6-2 Words and Phrases in a Sort Specification 6-11
7-1 List Processing Menu Options 7-2
7-2 Select Output Document Option Menu 7-5

About This Guide

PRODUCT: WPS-PLUS/ALL-IN-1 Version 1.0

FOR USE WITH: ALL-IN-1 Office Menu Version 2.0

Purpose of This Guide

WPS-PLUS/ALL-IN-1 List Processing explains how to use WPS-PLUS List Processing, List Processing Math, and Sort Processing.

List Processing and Sort Processing are features of WPS-PLUS, and WPS-PLUS is an optional feature of ALL-IN-1. For information about the WPS-PLUS Editor, see other books in the WPS-PLUS documentation set.

Audience

This guide is for users who want to produce documents such as form letters, mailing lists, and monthly financial reports.

Prerequisites

Before using List and Sort Processing, you should have completed the editing exercises in the *ALL-IN-1 Office Menu Getting Started Guide* or in *WPS-PLUS/ALL-IN-1 Getting Started.*

No previous List Processing experience is necessary.

Associated Documents

For details about how to use WPS-PLUS, you can also refer to the following documents:

WPS-PLUS/ALL-IN-1 Editor Functions

WPS-PLUS/ALL-IN-1 Getting Started

WPS-PLUS/ALL-IN-1 Installation

WPS-PLUS/ALL-IN-1 Quick Lookup

ALL-IN-1 Office Menu Getting Started Guide

ALL-IN-1 Office Menu: User's Reference

DECdx User's Guide

Conventions

ALL CAPS	Identifies a WPS-PLUS editor function. For example, press ADVANCE PARA means to press the ADVANCE key and then the PARA key.
Enter	Means type the letter(s) associated with a menu option and press RETURN.
<field name>	Text in angle brackets is a field name for List Processing or Sort Processing.
Gold X	Means press the Gold key, labeled PF1 on some keypads, then a function key. For example: Press Gold MENU.
(RETURN)	In an example, RETURN in parentheses indicates that the RETURN key was pressed.
WPS-PLUS	Means WPS-PLUS/ALL-IN-1.

1

List and Sort Processing

This chapter gives an overview of List Processing, List Processing Math, and Sort Processing.

What is List Processing?

List Processing merges lists of information with formatted text you create to produce form letters, reports, memos or other documents with a minimum of typing. You can produce letters or reports for everyone on a list or only for certain people, such as those living in New England. You can use List Processing Math to perform calculations on numeric values in the list.

You can use List Processing to:

- Create personalized letters describing a new product or service
- Produce a directory of customers
- Produce a telephone directory for a company
- Produce a list of purchase orders
- Produce a series of monthly reports in the same format

To use List Processing, you create:

- A list document

  The list document contains sets of related information, called records. For example, if you are producing individualized letters, your list document contains records composed of names and addresses.

- A form document

  The form document is a template for the letter or report. A form document shows how each copy looks when printed, except that the form document contains field names where values from the list document will be inserted. For example, if you are producing individualized letters, the form document contains the body of the letter and field names where names and addresses will appear.

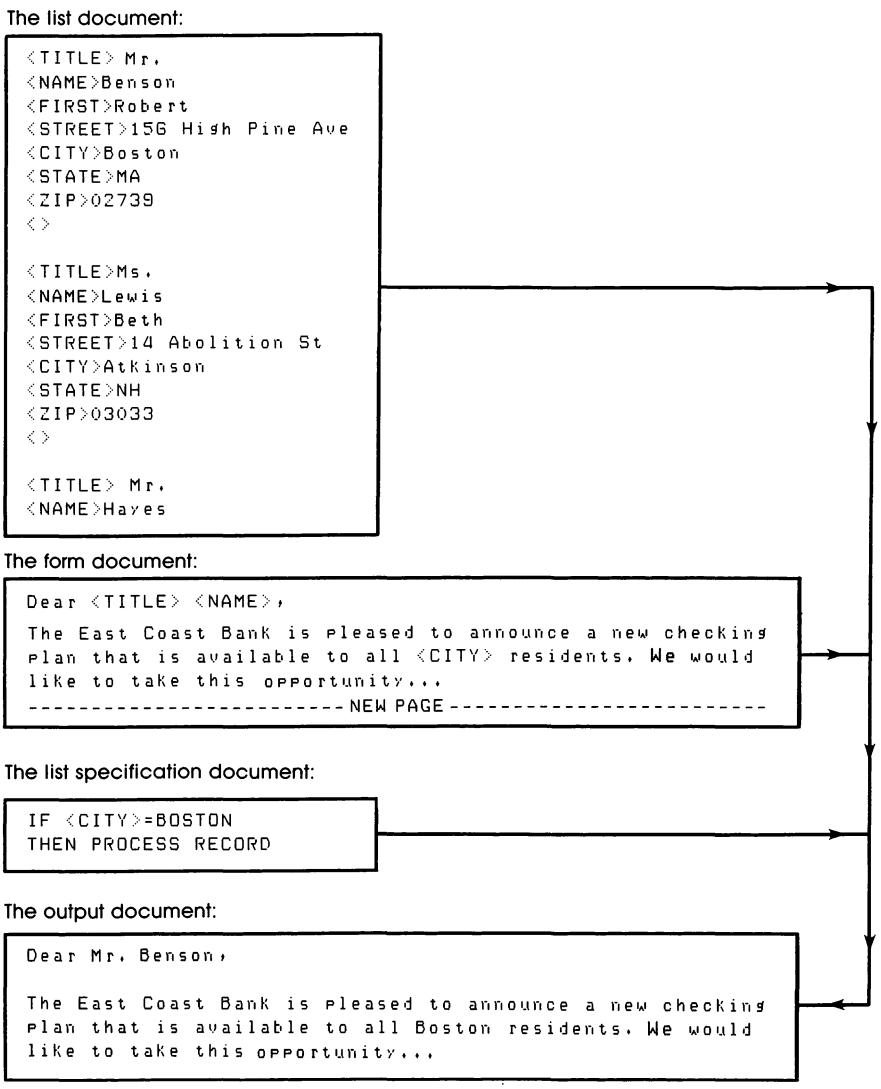
- A list specification document

  The list specification document tells List Processing which records to use from the list document.

List Processing uses your list document, form document, and list specification document to create an output document. For example, if you run List Processing to create forty individualized letters, the output document contains forty letters. You print or display the output document as you would any document.

How List Processing Works

Figure 1-1 shows how List Processing works.



MK-01661-00

Figure 1-1 How List Processing Works

The list document shown in Figure 1-1 contains records for Mr. Robert Benson and Ms. Beth Lewis. The form document contains a <TITLE> field, a <NAME> field, and a <CITY> field in a form letter. List Processing inserts a title, name, and city from a list document record into each copy of the form letter. The list specification document restricts the records processed to those with Boston in the <CITY> field. Thus, List Processing creates form letters for Boston residents only.

How to Create a List Document

To create a list document:

1. Enter C (Create) from the Word and Document Processing Menu and fill out the Creating New Document Form.

   When you press RETURN, the screen clears and you are in the editor.

2. Type a left angle bracket, <, the name for a field, and a right angle bracket, >.

   The right angle bracket completes the field name.

3. Type the value for that field immediately after the right angle bracket and press RETURN.

4. Repeat steps 2 and 3 until you have typed all field names and values for the record.

5. End the record with an end record marker, < >, by typing left and right angle brackets.

   You may add a blank line between records to make your list document easier to read.

6. Repeat steps 2 through 5 until you have typed all records in the list document.

7. Press Gold FILE DOCMT to file the list document.

Hints.

- Use the PARA key to move from one record to the next when editing the list document.
- Use the CUT function to store field names, then use the PASTE function as many times as there are records. After pasting field names for all records in your document, go back and fill in field values.
- Use the CUT and PASTE functions to remove records from the list and to move records from one place to another.
- Use the <> (ENTER key) on the keypad to advance or back up to the start of a field value.

NOTE: You can include multinational and technical characters in any document used in List or Sort Processing.

How to Create a Form Document

To create a form document:

1 Enter C (Create) from the Word and Document Processing Menu and fill out the Creating New Document Form.

When you press RETURN, the screen clears and you are in the editor.

2 Type the text that will go into the output document each time the template is used.

You may include rulers, bolding, and underlining.

3 Put field names where their field values should go in the output document. Type each field name exactly as it appears in the list document.

4 Press Gold NEW PAGE to end the form document if the form document is the template for a letter, report or label (if you want to start a new page each time the form document is processed).

Gold NEW PAGE inserts a NEW PAGE mark. The NEW PAGE mark will appear after each copy of the letter, report, envelope, or label in the output document. When you print the output document, each copy of the letter, report, envelope, or label will begin on a new page.

If the form document is the template for entries in a table, do not insert a NEW PAGE mark.

5 Press Gold FILE DOCMT to file the form document.

How to Create a List Specification Document

To create a list specification document:

1 Enter C (Create) from the Word and Document Processing Menu and fill out the Creating New Document Form.

When you press RETURN, the screen clears and you are in the editor.

2 Type PROCESS RECORD.

This list specification processes all records in your list document.

To process fewer records, use IF and one or more conditions as explained in Chapter 4.

3 Press Gold FILE DOCMT to file the list specification document.

How to Run List Processing

To run List Processing:

1 Enter LP (List Processing) from the Word and Document Processing Menu.

The List Processing Menu displays.

2 Enter DOC (Document Processing) from the List Processing Menu.

A form displays.

3 Follow the instructions at the bottom of your screen to run List Processing.

You can use the Printing Output Document (P) option from the List Processing Menu to print the output document.

For more information about running List Processing, see Chapter 7.

How to Get Help

Help is available on all List Processing menus and forms.

If you need Help on a menu or form, press Gold H. A window appears on your screen, and you follow the instructions displayed to go through the Help process.

If you need Help while editing, press Gold H. Follow the instructions on your screen to get Help for any key.

What is List Processing Math?

List Processing Math performs calculations on numerical information in the list document based on control commands you insert in the list document. List Processing Math can perform addition, +, subtraction, -, multiplication, *, and division, /.

You can use List Processing Math to:

- Produce weekly business reports for a company
- Select customers who are behind on payments, and produce letters asking for payment
- Produce annual sales reports
- Produce weekly payroll reports

To use List Processing Math:

Insert a FORMULA control block in a list document.

A FORMULA control block contains FORMULA commands. These commands tell List Processing Math which calculations to perform on field values in records after the control block. List Processing puts the results to calculations in your output document where indicated by the field names you put in the form document.

How List Processing Math Works

Figure 1-2 shows how List Processing Math works.

In Figure 1-2, FORMULA commands in the list document tell List Processing Math how to calculate the commission and profit for each sales representative. The form document tells List Processing where to write results in entries of the Sales Representative Report. The output document in Figure 1-2 shows how the results look in the report.

How to Create a FORMULA Control Block

To create a FORMULA control block in a list document:

1. Press Gold CMND to insert a START CONTROL marker.
2. Type WPSMATH.
3. Press RETURN.
4. Type FORMULA and press the space bar.
5. Type the formula you want List Processing Math to use.

   A formula includes the names of input fields separated by the arithmetic operators, +, -, *, or /. Input fields are followed by an equals sign, and the name of the result field. For example:

   ```
   FORMULA <:NUMBER OF ITEMS> * <:ITEM COST> = <:COST OF ITEMS>
   ```

   Input fields must be numeric fields that appear in list document records since List Processing Math gets input data from the list document. A left angle bracket and colon indicate a numeric field name.

   If your formula is too long for a single line, let it wrap to the next line.
6. Press the space bar.
7. Include a format string in quotation marks if you want results in a customized format.

   See Chapter 5.

The list document:

```
Sales Report

This is a list of all our sales representatives, their sales
for the month, their commissions, and their expenses.

---------------------- START CONTROL ----------------------
WPSMATH
FORMULA <:SALES> * 0.1 = <:COMMISSION> "$99,999.99"
FORMULA <:SALES> - <:COMMISSION> - <:EXPENSES> = <:PROFIT>
"$99,999.99"
----------------------- END CONTROL -----------------------

This list should be updated on the first monday of every
month and the report sent to Hill and Armstrong.

<NAME>Willie Logan
<AREA>New York
<:SALES>$8809.00
<:EXPENSES>$550.78
<>

<NAME>Yvonne Laddy
<AREA>London
<:SALES>$10,085.00
<:EXPENSES>$803.45
<>
```

The form document:

```
Sales Representative Report

<!S><NAME>, working in <AREA>, made <:SALES>. Expenses were
<:EXPENSES> and the commission was <:COMMISSION>, leaving a
profit of <:PROFIT>.
<!E>
```

The list specification document:

```
PROCESS RECORD
```

The output document:

```
Sales Representative Report

Willie Logan, working in New York, made $8809.00. Expenses
were $550.78 and the commission was $880.90, leaving a
profit of $7,377.32.

Yvonne Laddy, working in London, made $10,085.00. Expenses
were $803.45 and the commission was $1,008.50, leaving a
profit of $8,273.05.
```

MK-01662-00

Figure 1-2 How List Processing Math Works

8 Press RETURN.

When List Processing Math encounters a RETURN in a control block, it expects a new command or the end of the control block.

9 Include other FORMULA commands if you need them.

10 Press Gold CMND to insert an END CONTROL marker.

List Processing Math applies every FORMULA command in the control block to the records after the control block until it reaches another control block or the end of the list document.

What is Sort Processing?

Sort Processing arranges lists of alphabetic or numeric information in ascending or descending order. You use the sorted list as input for List Processing.

You can use Sort Processing to:

- Sort a mailing list by zip codes
- Alphabetize a file by sorting on customers' last names
- Maintain a filing system

Hint. You can add new entries to the beginning or the end of a list, then run Sort Processing to place the new entries in sequence with the old ones.

To use Sort Processing, you create:

- A list document

  The list document contains the records to be sorted. You can sort a list document you have created for List Processing.

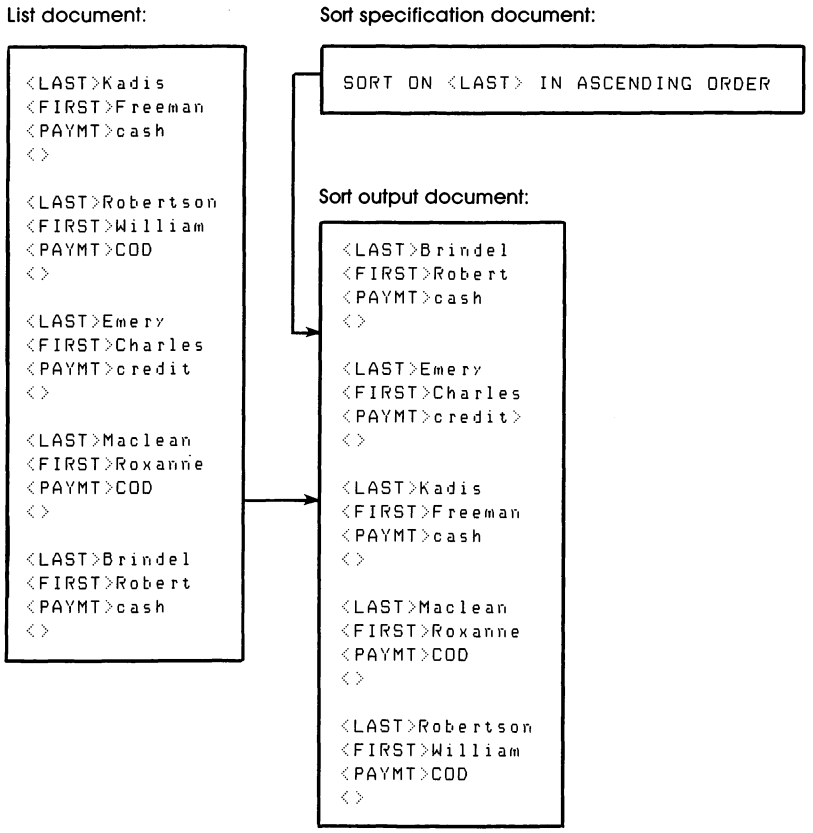
- A sort specification document

  The sort specification document tells Sort Processing which record field to use in sorting records, and which order to sort in (ascending alphabetic/numeric order or descending alphabetic/numeric order).

When you run Sort Processing, you produce a sort output document. The sort output document is a sorted list.

How Sort Processing Works

Figure 1-3 shows how Sort Processing works.



MK-01663-00

Figure 1-3 How Sort Processing Works

In Figure 1-3, the sort specification document sorts records by the <LAST> field, which contains last names, and in ascending order. Ascending order is the same as alphabetical order when letters are compared. Thus, Sort Processing arranges the list document records in alphabetical order by the last names in the records.

How to Create a Sort Specification Document

To create a sort specification document:

1. Enter C (Create) from the Word and Document Processing Menu and fill out the Creating New Document Form.

   When you press RETURN, the screen clears and you are in the editor.

2. Type SORT ON and a space.
3. Type the name of the field you are using to sort records and press the space bar.

   To sort on more than one field, see Chapter 6.

4. Type IN ASCENDING ORDER to sort in ascending alphabetic/numeric order.

   To sort in descending order, type IN DESCENDING ORDER.

5. Press Gold FILE DOCMT to file the sort specification document.

How to Run Sort Processing

To use Sort Processing:

1. Enter LP (List Processing) from the Word and Document Processing Menu.

   The List Processing Menu displays.

2. Enter SR (Sort Processing) from the List Processing Menu.

   A form displays.

3. Follow the instructions at the bottom of your screen to run Sort Processing.

For more information about running Sort Processing, see Chapter 7.

2

Preparing Your Lists

This chapter explains:

- How to type field names
- How to type field values
- How to control list document size
- How to put list document records in order
- How to create numeric fields
- How to prevent gaps in your data and output
- How to insert comments in a list document
- How to use abbreviation and library documents

List Documents

A list document provides the raw information for List Processing or Sort Processing.

For steps to follow in creating a list document, see Chapter 1.

Records

A list document consists of records. A record consists of the fields between two end record markers (<>) or all fields before the first end record marker.

Fields

Each field consists of a field name followed by a field value:

```
<field name>field value
```

The following example shows fields in a record for a mailing list:

```
<TITLE>Ms.
<FIRST>Debra
<LAST>Smith
<ADDRESS>12 Hill Street
Grantham,
Lincolnshire,
GR1 0bt
England
<>
```

Typing Field Names

Field names have the following characteristics:

- A field name starts with a left angle bracket, <, and ends with a right angle bracket, >.
- A field name cannot contain additional angle brackets.

- A field name cannot contain exclamation points, !. An exclamation point inside angle brackets indicates a command. See Chapter 3.
- A field name can have uppercase or lowercase letters. To List Processing, <ADDRESS>, <Address>, and <address> are the same field name.
- Field names in the list, form, and list specification documents must match. For example, if you use <ADDRESS> in your list document, you cannot use <ADDR> in your form document or your list specification document to refer to the same field.

Typing Field Values

Make sure you type field values in a list document exactly as you want them to appear in your output. For example, do not use uppercase letters only unless you want your output to be in uppercase.

You can put RETURNs in field values. In the previous example, 12 Hill Street was separated from Grantham by a RETURN. When List Processing inserts the value of <ADDRESS> into the output document, it inserts a RETURN after Street so Grantham is on a separate line.

Do not type a space or a tab before a field value, as it can cause unexpected results. If you type a tab in front of a street address field value, your output document will print like this:

```
Ms. Debra Smith
        12 Hill Street
Grantham,
Lincolnshire,
GR1 0bt
England
```

List Document Guidelines

The following sections contain helpful guidelines for creating a list document.

List Size

You can break up a very long list document into shorter ones, which are easier to use. For example, instead of creating one list document for all customers, create one list document for customers whose last names begin with letters A through D, another document for customers E through H, and so on.

Order of Records

If the order of the records in the list document is important, insert each new record in the correct location or use Sort Processing to arrange your list. List Processing processes records in the order of their appearance in the list document.

Order of Fields

You can arrange record fields in a list document in any order. However, lists are easier to maintain if you enter fields in the same order for each record.

Using Numeric Fields

You can use List Processing Math to perform calculations on values in numeric fields. A numeric field name is a field name that begins with a left angle bracket and colon, <:. For example:

```
<:CREDIT>
```

For more information about numeric fields, see Chapters 4 and 5.

Entering Dates as Field Values

See Chapter 4.

Missing Fields and Field Values

Be careful not to accidently omit a field or a field value from a record. If a field value, or both a field name and field value, are missing from a record, List Processing cannot insert the value in the output document. List Processing does, however, insert a blank where the value would go.

Inserting Comments

You can put comments in a list document. Comments help others who use your list by explaining what your list contains or how to process the list.

Here is an example of a useful comment:

```
---------------------- START CONTROL------------------------
COMMENT

This is the customer mailing list CUSTN1 for customers A-M.  Each
record has fields <TITLE>, <FIRST>, and <LAST> for a customer name,
and <ADDRESS> for a customers complete mailing address.  The <:OWED>
field contains the amount a customer owes us at the end of each
month. See document CUSTN2 for customers N-Z.
---------------------- END   CONTROL -------------------------
```

Include a comment in a control block if:

- The comment must be easy to find.
- The comment contains a field name.
- The comment is between the first field in a record and the end record marker.

To put a Comment in a control block:

1 Press Gold CMND to start the control block.

2 Type COMMENT and press RETURN.

3 Type the comment.

4 Press Gold CMND to end the control block.

To include a comment without using a control block:

Simply type the comment.

Comments do not appear in the output document unless you are using the <!R> command to copy a list document record, and the comment is in the record.

Using Abbreviation and Library Documents

If you use List Processing often, store blank records in an abbreviation or library document. For information about abbreviation and library documents, see *WPS-PLUS Editor Functions*.

3

Formatting Your Output

This chapter explains how to:

- Create form documents to produce different types of output
- Put leaders and trailers in an output document
- Single space, double space, or group table entries in an output document
- Skip processing list document records with a form document command
- Copy all or part of a list document
- Change field names in list document records with a form document command

Form Documents

A form document is a template that tells List Processing:

- Which field values from a list document to insert in the output document and where to insert them
- Which information, if any, to insert in the output document in addition to field values from the list document

For example, if you are creating form letters, your form document contains a copy of the form letter and field names from the list document.

If you are creating address labels, your form document contains only field names from the list document.

For the steps to follow in creating form documents, see Chapter 1.

Types of Form Documents

The following sections explain how to create form documents for:

- Continuous-form letterhead
- Continuous-form envelopes
- Continuous-form labels
- Tables
- Tables with more than one column

NOTE: After you run List Processing, print letters, envelopes, or labels on a letter-quality printer.

Set the Total lines per page print setting to the number of lines between the top of one letter, envelope, or label and the next. Take into account the Vertical lines per inch print setting. For example, if the Vertical lines per inch setting is eight, an inch of printed text contains eight lines.

If you are putting headers or footers on your output, make sure the value for the Current text size setting is correct. See WPS-PLUS Editor Functions for information about settings.

Continuous-Form Letterhead

Letterhead attached to continuous-form paper is called continuous-form letterhead.

To format a form document for continuous-form letterhead:

1. Create a ruler that contains a left margin (L) where you want the salutation to start. Put a tab (T) in the ruler where you want the return address and closing of each letter to start.
2. Press RETURN to move the cursor to the first line of the return address.
3. Press TAB to move the cursor to the start of the return address.
4. Type the return address.
5. Press RETURN to move the cursor to the first line of the salutation.
6. Type the salutation, followed by the field names <TITLE> and <LAST>.

   Put a space between two successive fields on the same line. For example:

   ```
   Dear <TITLE> <LAST>,
   ```

7. Type the text of the form letter, with the desired field names.

   Each field name in the form document must match a field name in the list document.

   You do not need to press RETURN after the last line of text.

8. Press Gold NEW PAGE to insert a NEW PAGE mark.
9. Press Gold FILE DOCMT to file your document.

For example:

```
L-----------------------------------T-------------------------------R


                                    Southport Bookstore
                                    100 Beacon St.
                                    Southport, NH   03039

Dear <TITLE> <LAST>,

All of us at the Southport Bookstore would like to thank you, <TITLE>
<LAST>, for your patronage this year.

                                    Sincerely,

                                    William Howard
------------------------------ NEW PAGE -------------------------------
```

For each list document record processed with this form document, List Processing inserts a different title and last name into the letter.

Here are sample list document records:

```
<TITLE>Mrs.
<FIRST>Deborah
<LAST>Howe
<STREET>444 Evans Road
<CITY>Londonderry
<STATE>NH
<ZIP>03053
<>

<TITLE>Mrs.
<FIRST>Beth
<LAST>Morris
<STREET>737 Poplar Avenue
<CITY>Londonderry
<STATE>NH
<ZIP>03053
<>
```

Your output document contains this information after processing:

```
                                    Southport Bookstore
                                    100 Beacon St.
                                    Southport, NH   03039

Dear Mrs. Howe,

All of us at the Southport Bookstore would like to thank you, Mrs.
Howe, for your patronage this year.

                                    Sincerely,

                                    William Howard
------------------------------ NEW PAGE ---------------------------



                                    Southport Bookstore
                                    100 Beacon St.
                                    Southport, NH   03039

Dear Mrs. Morris,

All of us at the Southport Bookstore would like to thank you, Mrs.
Morris, for your patronage this year.

                                    Sincerely,

                                    William Howard
------------------------------ NEW PAGE ---------------------------
```

If your form document has more than one page, use any of the methods discussed in *WPS-PLUS Editor Functions* to paginate your form document. End the form document with a NEW PAGE mark.

Continuous-Form Envelopes

Envelopes attached to continuous-form paper or attached end-to-end are called continuous-form envelopes.

To format a form document for continuous-form envelopes:

1. Create a ruler with left margin (L) where you want the return address (if desired) and a tab (T) where you want the address from the list document to start.

2 Press RETURN to move the cursor to the first line of the return address.

3 Type the return address.

4 Press RETURN to move the cursor to the first line of the address.

5 Tab to where the address should go.

6 Type the field names for name and address.

Put a space between two successive fields on the same line.

You do not need to press RETURN after the last line of text.

7 Press Gold NEW PAGE to insert a NEW PAGE mark.

8 Press Gold FILE DOCMT to file your document.

For example:

```
L---------------------------------------T----------------------R

Southport Bookstore
100 Beacon St.
Southport, NH   03039
                                        <TITLE> <FIRST> <LAST>
                                        <STREET>
                                        <CITY>, <STATE>   <ZIP>
------------------------------ NEW PAGE --------------------------
```

Your output document contains this information after processing:

```
Southport Bookstore
100 Beacon St.
Southport, NH   03039
                                        Mrs. Deborah Howe
                                        444 Evans Road
                                        Londonderry, NH   03053
------------------------------ NEW PAGE --------------------------
```

(continued)

```
Southport Bookstore
100 Beacon St.
Southport, NH   03039
                                        Mrs. Beth Morris
                                        737 Poplar Avenue
                                        Londonderry, NH   03053
------------------------------ NEW PAGE -----------------------
```

Continuous-Form Labels

To format a form document for continuous-form labels:

1. Create a ruler with right and left margins (R and L).
2. Move the cursor to where the text on the label should start.
3. Enter the field names for title, first name, and last name.
4. Enter the field name(s) for the address.
5. Press Gold NEW PAGE to begin the next label.
6. Press Gold FILE DOCMT to file your document.

For example:

```
L------T-----------------------------------------R--------------

       <TITLE> <FIRST> <LAST>
       <STREET>
       <CITY>, <STATE>   <ZIP>
------------------------------ NEW PAGE --------------------------
```

Your output document contains this information after processing:

```
       Mrs. Deborah Howe
       444 Evans Road
       Londonderry, NH   03053
------------------------------ NEW PAGE --------------------------

       Mrs. Beth Morris
       737 Poplar Avenue
       Londonderry, NH   03053
------------------------------ NEW PAGE --------------------------
```

Tables

If your document is a single list or table, you need to print your output continuously down each page until the end of the table is reached.

To print your output continuously:

Omit NEW PAGE marks in the form document.

For example, you can create a mailing list from a list document of customer records, and a form document containing only the field names that appear in the mailing list. Suppose each list document record contains the following fields:

```
<FIRST>
<MIDDLE>
<LAST>
<STREET>
<CITY>
<STATE>
<ZIP>
<TELEPHONE>
<CREDIT>
<LAST UPDATED>
```

Your form document contains the fields that appear in mailing list entries:

```
<LAST>, <FIRST>
<STREET>
<CITY>, <STATE> <ZIP>(RETURN)
```

Your output document is a mailing list in which blank lines, inserted by the last RETURN in the form document, separate records.

For example:

```
Collins, Nancy
649 Hawthorne Road
Bloomington, IN   47041

Dumas, Andre
19 Walnut Street
Bloomington, IN   47041

Roberts, William
30 First Street
Bloomington, IN   47041
```

Form Document Commands

Form document commands change the formatting of your output document. Table 3-1 summarizes the commands you use in a form document.

Table 3-1 Form Document Commands

Use	To
<!S>	Include leader text to be printed at the top of your output document.
<!E>	Include trailer text to be printed at the bottom of your output document.
<>	Advance to the next list document record.
<!R>	Copy records to another list document.

The following sections explain how to use these commands.

Leaders and Trailers

When creating a table or directory, you may want to put information before or after the list. If your list document is a list of parts, you may want a title and column headings to precede the list of parts. Put the title and column headings in a leader. You may also want to include ordering instructions after the list. Put the ordering instructions in a trailer.

A leader is information inserted at the top of your output document.

A trailer is information inserted at the bottom of your output document.

In the form document, the area between the leader and the trailer is the repetitive area.

List Processing:

- Processes the leader once, copying the leader into the output document
- Processes list document records, using the repetitive area of the form document as many times as needed until there are no more records to process
- Processes the trailer once, copying the trailer into the output document

To put a leader in your output document:

1 Type the text for the leader at the top of your form document.

2 Type the <!S> command on a separate line after the leader.

The <!S> command starts the repetitive area.

To put a trailer in your output document:

1 Type the <!E> command at the bottom of the repetitive area in your form document (after the part of your form document that is processed repeatedly).

The <!E> command ends the repetitive area.

2 Type the text for the trailer after the <!E> command.

Here is a sample list document:

```
<PART#>80-0030
<NAME>Lockwasher
<:COST>0.015
<>

<PART#>80-0040
<NAME>Diode
<:COST>0.02
<>

<PART#>80-0235
<NAME>Glue
<:COST>0.01
<>
```

Here is a form document with <!S> and <!E> commands, a leader, and a trailer:

```
L--------T---------C--------------------,-------------R

                PARTS LIST
               for Project X

Part      Part Name                 Cost
Number    ---------                 ----
<!S>
<PART#>   <NAME>                    <:COST><!E>

Parts can be ordered from Stockroom 5.
```

Here is sample output showing the leader and trailer:

```
              PARTS LIST
             for Project X

Part      Part Name                      Cost
Number    ---------                      ----

80-0030   Lockwasher                     0.015
80-0040   Diode                          0.02
80-0235   Glue                           0.01

Parts can be ordered from Stockroom 5.
```

When processing starts, the leader is written to the output document. Then processing of the repetitive area of the form document begins.

The repetitive area consists of one line which tells List Processing to retrieve the values of the fields named and place them in the output document. Values are typed at the tab positions where field names appear.

After processing all records in the list document, List Processing copies the trailer into the output document and processing is completed.

Spacing Leaders and Trailers

If the number of blank lines after the leader or before the trailer is important, check your output document before printing.

To put more blank lines after the leader:

Add blank lines between the leader and the <!S> command.

To put fewer blank lines after the leader:

Remove blank lines between the leader and the <!S> command.

To put more blank lines before the trailer in your output document:

Add blank lines between the <!E> command and the trailer.

To put fewer blank lines before the trailer:

Remove blank lines between the <!E> command and the trailer.

NOTE: RETURNs at the end of the repetitive area also affect the spacing before the trailer in your output document.

Spacing Table Entries

The next sections explain how to:

- Single space table entries in the output document
- Double space table entries in the output document
- Group table entries in the output document

Single and Double Spacing

To single space table entries if your form document does not contain an <!S> or an <!E> command, simply:

End your form document without pressing RETURN.

For example:

```
L--------T--------------------,-----------------------R
<PART#>  <NAME>           <:COST>
```

Each time List Processing begins processing the repetitive area, a table entry is started on a new line, so table entries are single-spaced. For example:

```
80-0030  Lockwasher            0.015
80-0040  Diode                 0.02
80-0235  Glue                  0.01
```

To double space entries if your form document does not contain an <!S> or an <!E> command:

Press RETURN at the end of your form document.

For example:

```
L--------T--------------------,-----------------------R
<PART#>  <NAME>           <:COST>(RETURN)
```

Here is sample output:

```
80-0030   Lockwasher          0,015

80-0040   Diode               0,02

80-0235   Glue                0,01
```

The <!S> and <!E> commands can affect the spacing of table entries in your output document. Table 3-2 explains how to single space or double space table entries when you are using the <!S> or <!E> command or both.

Table 3-2 Single and Double Spacing with <!S> and <!E>

If the Form Document Contains	To Single Space Entries	To Double Space Entries
<!S> only	Put <!S> immediately before the first record field. End the repetitive area without a RETURN.	Put <!S> on a separate line. End the repetitive area without a RETURN.
<!E> only	Put <!E> on a separate line.	Put a blank line before <!E>.
<!S> and <!E>	Put <!S> on a separate line. Put <!E> immediately after the last record field.	Put <!S> on a separate line. Put <!E> on a separate line.

Grouping Entries

You can group the output from several list document records. Here is a sample list document:

```
<PART#>80-0030
<NAME>Lockwasher
<:COST>0.015
<>

<PART#>80-0040
<NAME>Diode
<:COST>0.02
<>

<PART#>80-0235
<NAME>Glue
<:COST>0.01
<>

<PART#>80-0451
<NAME>Hex Screw
<:COST>0.03
<>

<PART#>80-0477
<NAME>Hex Nut
<:COST>0.017
<>
```

Here is a sample output document:

```
                PARTS LIST
               for Project X


Part       Part Name                        Cost
Number     ---------                        ----

80-0030    Lockwasher                       0.015
80-0040    Diode                            0.02
80-0235    Glue                             0.01

80-0451    Hex Screw                        0.03
80-0477    Hex Nut                          0.017

Parts can be ordered from Stockroom 5.
```

The field values for Lockwasher, Diode, and Glue are in different list document records. <> commands in the form document group the output from three records.

Here is the form document:

```
L--------T---------C--------------------,------------R
                PARTS LIST
               for Project X

Part      Part Name                 Cost
Number    ---------                 ----
<!S>
<PART#>   <NAME>                    <:COST><>
<PART#>   <NAME>                    <:COST><>
<PART#>   <NAME>                    <:COST>
<!E>

Parts can be ordered from Stockroom 5.
```

A <> command begins processing of the next list document record immediately. Thus three list document records are processed each time the repetitive area is processed. The placement of the <!S> and <!E> commands puts a blank line after every three lines of output instead of after each line.

To group the output from several records:

1. Use the form document to group your output.
2. Use <> commands to control when records are processed.

Columns

To create an output document with columns:

1. Put tab stops in your form document ruler.
2. Use the <> command between tab stops to get the next record.

Here is a list document:

```
<ITEM>apple
<:COST>.27
<>
<ITEM>fig
<:COST>.05
<>
<ITEM>pineapple
<:COST>1.36
<>
<ITEM>pear
<:COST>.32
<>
<ITEM>banana
<:COST>.08
<>
<ITEM>plum
<:COST>.16
<>
```

Here is a form document:

```
L-------------------,--------------T-------------------,--------R
<ITEM>         <:COST><>             <ITEM>         <:COST>
```

List Processing:

- Processes the first record and puts values at the left margin and the first decimal tab stop
- Processes the <> command and gets the next record
- Processes the record and puts information at the other tab stops
- Returns to the "top" of the form document, the left margin, gets the next record and begins the next line of the output document

List Processing outputs records row by row. For example:

```
apple                .27            fig                  .05
pineapple           1.36            pear                 .32
banana               .08            plum                 .16
```

Skipping Records

To skip processing a list document record:

Put a <> command in the form document at the end of the repetitive area.

For example:

```
L--------T---------C--------------------,-------------R
              PARTS LIST
             for Project X

Part      Part Name                 Cost
Number    ---------                 ----
<!S>
<PART#>   <NAME>                    <:COST><><!E>


Parts can be ordered from Stockroom 5.
```

The <> command between <:COST> and <!E> sends processing to the next record. In addition, the end of the repetitive area sends processing to the next record after that. Thus, only every other record is processed.

Copying Lists

To copy a list document:

Use the <!R> command in a form document.

You can include comments for the new list document as a leader or trailer in the form document. Here is a sample form document that produces a list document as output. The <!R> command inserts an exact copy of each record in the output:

```
This file contains active customer accounts.

<!S><!R><!E>

Parts can be ordered from Stockroom 5.
```

Use the <!R> command by itself if you want blank lines between records.

Hints.

- Use the <!R> command with different selection specifications to produce sublists from a longer list
- Use the <!R> command to produce many copies of a list document
- Use Gold GET DOCMT in the WPS-PLUS Editor to produce one copy of a list document

Changing List Document Field Names

You can change the field names in all records in a list document by running List Processing to create a new list document.

To change field names:

1. Create a form document containing the new field names.
2. Put old field names in the form document where field values will go in the new list document.
3. Run List Processing to create a new list document as your output document.

Hint. You can use this procedure to remove certain fields from all list document records or to add fields to all list documents.

Here is a sample record from a list document:

```
<NAME>
<POSITION>
<LOCATION>
<MAIL ADDRESS>
<TELEPHONE>
<>
```

Suppose you want the list document to have fewer fields and shorter field names. It may take a long time to retype the list document. In addition, retyping a document usually introduces errors.

To change the field names in the list document, create a form document containing the new field names. List Processing must interpret the left and right angle brackets of the field names literally and write them into the output document. However, List Processing usually interprets field names in the form document as places to insert field values. The left and right angle brackets do not show up in the output document.

To put a left angle bracket in your output document:

Put two left angle brackets, <<, in the form document.

A single left angle bracket is interpreted as the start of a field name or a command.

To put a right angle bracket, >, in your output document:

Put a single right angle bracket in the form document.

To change the field names in the list document shown above, create a form document such as:

```
<<NA><NAME>
<<LOC><LOCATION>
<<PHONE><TELEPHONE>
<<>
```

Here is a sample record from the output document.

```
<NA>Arthur Mitchell
<LOC>Eastern Region Sales Office
<PHONE>838-4853
<>
```

4

Selecting the Records to Process

This chapter explains how to create list specification documents to:

- Process only the list document records that contain a certain value or values
- Process only the records that contain one of several values
- Reject records that contain a certain value or values
- Test the numeric value, the characters, or the digits in a field
- Test a list specification document without running List Processing

List Specification Documents

A list specification document tells List Processing which records in a list document to process. You can process all records in a list document, or you can process only certain records. For example, if you are processing a mailing list to produce address labels, you can process records only for people who live in certain cities or states.

For the steps to follow in creating a list specification document that processes all records, see Chapter 1.

Restricting the Records You Process

To restrict the records that are processed:

Include an IF-statement in your list specification document.

An IF-statement is a list of conditions that begins with IF and ends with THEN PROCESS RECORD. THEN PROCESS RECORD must be on a separate line.

The simplest condition is an IF-condition. An IF-condition has the form:

```
field name=field value
```

This type of condition is always the first one in an IF-statement.

For example:

```
IF <STATE>=NH
THEN PROCESS RECORD
```

List Processing checks each list document record to see if the <STATE> field in the record has the field value NH.

Here is a record that is selected and processed:

```
<LAST>Davis
<FIRST>Sandra
<STREET>Poplar Street
<CITY>Londonderry
<STATE>NH
<ZIP>03053
```

There are three other conditions you can use to select records:

- OR-conditions
- AND-conditions
- BUT NOT IF conditions

Testing For Any of Several Values

To examine a field for any of two or more values:

Use one or more OR-conditions.

An OR-condition has the form:

```
OR=field value
```

For example:

```
IF <STATE>=New Hampshire
    OR=NH
    OR=N,H,
THEN PROCESS RECORD
```

This example selects records referring to New Hampshire, accepting any of three values for the field <STATE>.

You can use OR to modify AND or BUT NOT IF as well as IF. See the next sections.

Testing For More Than One Value

To examine more than one field in a record for desired values:

Use an AND-condition.

An AND-condition has the form:

```
AND field name=field value
```

In the next example, the list specification document uses AND to find records for residents of Hudson, New Hampshire:

```
IF <CITY>=Hudson
    AND <STATE>=New Hampshire
THEN PROCESS RECORD
```

List Processing selects a record only if field values in the record match the values before and after AND.

You can use OR to modify AND. For example:

```
IF <CITY>=Hudson
    AND <STATE>=New Hampshire
    OR=NH
THEN PROCESS RECORD
```

List Processing selects records for residents of Hudson, New Hampshire and accepts two values for New Hampshire.

NOTE: Do not insert a space or a tab either between the equals sign and the field value, or after the field value. Accidental characters can be hard to find later and they can prevent correct processing of your list document.

Rejecting Records

To reject some records among those you are selecting:

Use a BUT NOT IF condition.

A BUT NOT IF condition has the form:

```
BUT NOT IF field name=field value
```

The next example leaves out certain records even though they pertain to residents of Manchester, New Hampshire:

```
IF <CITY>=Manchester
AND <STATE>=New Hampshire
    OR=NH
BUT NOT IF <CREDIT>=BAD
THEN PROCESS RECORD
```

You can use OR to modify BUT NOT IF. For example:

```
IF <CITY>=Manchester
AND <STATE>=New Hampshire
    OR=NH
BUT NOT IF <CREDIT>=BAD
    OR=NO RATING
THEN PROCESS RECORD
```

NOTE: If you put AND after BUT NOT IF, AND modifies the IF at the top of the IF-statement. If you use two successive BUT NOT IFs, both modify the IF at the beginning of the IF-statement.

Using More Than One IF-Statement

To use more than one IF-statement in a list specification document:

Link IF-statements with OR.

The next list specification document consists of three IF-statements. Blank lines between IF-statements show the proper grouping of ideas:

```
IF <STATE>=Maine
THEN PROCESS RECORD

OR IF <STATE>=Massachusetts
    AND <COUNTY>=Middlesex
THEN PROCESS RECORD

OR IF <STATE>=New Hampshire
    AND <COUNTY>=Hillsborough
        OR=Rockingham
        OR=Strafford
THEN PROCESS RECORD
```

Each record that satisfies at least one IF-statement is selected. The list specification document selects records that pertain to Maine and to counties in two other states.

Testing for Empty Fields

If only a RETURN follows the equals sign in an IF-statement, List Processing selects only the records that have no values for the field you identify. By testing for blank fields, you can locate fields you need to update.

To select records in which there is no field value for a field:

Test the field by putting nothing after the equals sign.

The next list specification document selects records with an empty <ZIP> field value:

```
IF <ZIP>=(RETURN)
THEN PROCESS RECORD
```

Suppose your records have an <HOURLY> field that contains a YES or NO value. With one list specification document, you can select the YES records. With another, you can select the NO records. With a third, you can select records with an empty <HOURLY> field value.

NOTE: List Processing does not select records that are missing the field name given in your IF-statement. If List Processing does not find the specified field name in a record, the record is not processed.

Testing with Wild Cards

There are two special characters, called wild cards, you can use in a list specification:

- The <?> (question mark) wild card

  To test for any single character:

  Use the <?> wild card.

  The next list specification document tells List Processing to select all records whose <CREDIT> fields contain exactly one character. Records with <CREDIT> field values that are empty or missing, or records with more than one character, are not selected:

  ```
  IF <CREDIT>=<?>
  THEN PROCESS RECORD
  ```

  You can combine the <?> wild card with partial field values. List Processing looks for the characters you specify and for any character where you used <?>. For example:

  ```
  IF <MODEL#>=R<?>-15<?>
  THEN PROCESS RECORD
  ```

List Processing looks in <MODEL#> fields for a capital R, any single character, a hyphen, a 1, a 5, and then any character, in that order. For example, RA-156 and R3-15B are two model numbers that are selected for processing.

To look for a certain number of characters, which can be any characters at all:

Type one question mark in angle brackets for each character.

The next list specification document searches for field values with exactly five characters at the start, before the letters *b-o-r-o*:

```
IF <TOWN>=<?????>boro
THEN PROCESS RECORD
```

Fields named <TOWN> with values Northboro and Southboro match this selection. Westboro (only four characters before *boro*) does not.

- The <*> (asterisk) wild card.

To look for any number of characters, which can be any characters at all, including none:

Use the <*> wildcard.

In the next example, any <TOWN> field value matches this specification if the last four characters are *boro*. Missing and empty field values do not match:

```
IF <TOWN>=<*>boro
THEN PROCESS RECORD
```

Values such as Northboro, Southboro, Eastboro, and Westboro match. A field containing only the four characters *boro* also matches.

The next list specification document selects any record where the <NAME> field value contains Jones, regardless of the characters that precede or follow it:

```
IF <NAME>=<*>Jones<*>
THEN PROCESS RECORD
```

Values such as Jones, Mary Jones, J.P. Jones, Sr., or John Paul Jones, Jr., all match this list specification.

The next list specification document selects any record that contains a <ZIP> field name, even if the record does not contain a <ZIP> field value:

```
IF <ZIP>=<*>
THEN PROCESS RECORD
```

Testing Field Values in Lists

In a list specification document, you can test field values:

- Numerically
- Character-for-character
- By their digits only

The next sections explain how to perform these tests.

Comparing Numeric Fields

List Processing tests the numeric value of a field if:

- The field name is numeric, such as <:COST>.
- The value on the right side of the equals sign is in angle brackets, such as <$200>.

The next example shows how to test the value of a numeric field:

```
IF <:COST>=<$200>
THEN PROCESS RECORD
```

When you test the value of a number, List Processing:

- Ignores leading zeros, asterisks, plus signs, commas, and dollar signs in the field value. Thus, 10000, 010000, *10000, +10000, 10,000, and $10000 are all equal.

- Interprets parentheses around a number, or leading or trailing minus signs, as indicating a negative value. Thus, (10000), -10000 and 10000- are equal values. But these numbers have a value different from 10000.
- Recognizes decimal points. Thus 10000 and 10000.00 are equal, but 1000.0, and 100.00 are not equal.

Comparing Character-For-Character

List Processing tests a field value character-for-character if the value on the right of the equals sign is not enclosed in angle brackets, such as $200. The field name can be numeric or nonnumeric.

Many of the examples in this chapter use character-for-character comparison. Make sure you use this method only when you need it. For example, suppose you wanted to select records for projects that cost $10,000:

```
IF <:COST>=$10,000
THEN PROCESS RECORD
```

List Processing looks for a character-for-character match. Records without the dollar sign or comma are not selected.

Comparing Digits Only

List Processing tests only the digits in a field if:

- The field name is not numeric.
- The value on the right of the equals sign is in angle brackets, such as <COST> = <$100>. In this example, List Processing tests only the characters 1, 0, and 0. The other character in this example, $, is ignored.

The next example shows how to test digits only:

```
IF <COST>=<10000>
THEN PROCESS RECORD
```

The field values $10,000, $10000, 10000, $100.00, 10,000, and 100.00 for <COST> match because each contains the digits 10000.

NOTE: You cannot use the <> and <?> wildcards on a line where you are using the angle bracket method to check for a number or numeric range.*

Testing Against Numeric Ranges

In any condition in an IF-statement, you can test if the value in a numeric field falls in a range. List Processing compares the field values in the list document record field you identify with the range on the right of the equals sign.

To test with a range, use one of these formats:

```
IF <:field name>=<field value> OR MORE

    IF <:ITEM#>=<10> OR MORE


IF <:field name>=<field value> OR LESS

    IF <:COST>=<$9.00> OR LESS


IF <:field name>=<field value 1> THROUGH <field value 2>

    IF <:INVENTORY>=<51> THROUGH <100>
```

Make sure to put field values in angle brackets when you test with a range.

The next example shows how to select records for employees whose salaries fall in a range:

```
IF <:SALARY>=<10000> THROUGH <14999>
THEN PROCESS RECORD
```

NOTE: You can use THRU for THROUGH.

You can select customers who have owed $50 or more for at least three months:

```
IF <:BALANCE>=<$50.00> OR MORE
AND <:OVERDUE>=<3> OR MORE
THEN PROCESS RECORD
```

If you use a nonnumeric field name on the left of the equals sign, List Processing tests digits only, because numbers on the right of the equals sign are in angle brackets. (See the section Comparing Digits Only.) Thus, you do not need to include any characters except digits in your test values.

For example:

```
IF <BALANCE>=<50> OR MORE
AND <OVERDUE>=<3> OR MORE
THEN PROCESS RECORD
```

Testing Dates

If you will not use ranges to test dates, you can enter dates in your list document using the easy method explained in this section.

If you need to be able to test dates using ranges, see the next section.

To test dates in a list specification document:

1. Enter the day and month portions of all dates in your list document and list specification document as two-digit numerals. For example: 01/03/81.
2. Test digits only.

Entering each part of a date as a two-digit number ensures that values for one part of a date are compared with values for the same part. Unless each part of a date is entered as two digits, List Processing considers dates such as these to be the same:

```
1/23/81
12/3/81
```

Testing only the digits and not the characters separating the digits ensures that List Processing selects dates entered with dashes and slashes. Thus, List Processing selects dates such as 01-06-84 and 01/07/84.

Testing Dates Against a Range

To test if dates fall in a numeric range:

1. Put dates in numeric fields in your list document.
2. Enter dates without hyphens or slashes.

   Numeric fields cannot accept hyphens or slashes.
3. Enter all dates in the format YYMMDD (years, months, days).

   If you enter dates in the usual way, a date such as January 8, 1984 tests out larger than January 7, 1986 (because 010884 is larger than 010786).

For example:

```
IF <:DATE>=<410714> THROUGH <780621>
THEN PROCESS RECORD
```

This list specification tests for dates that fall between July 14, 1941 and June 21, 1978.

List Specification Guidelines

Follow these guidelines when you create a list specification document:

- Use as many characters or lines in the list specification document as you wish.
- Use spaces or tabs to indent lines of a list specification document where necessary. These spaces and tabs make no difference to the instructions, but they make instructions more readable.
- Type each field name exactly as it appears in list and form document records.

- Type each field value immediately after an equals sign. Put spaces or tabs in a value only if they also appear in the field values in the list document. Uppercase letters match only uppercase letters. Underlining and bolding are not considered.
- Put each field value or range on a separate line.
- When you type the words THEN PROCESS RECORD, put them on a separate line (after a RETURN).

Testing a List Specification Document

Use the Test Specification Document (T) option on the List Processing Menu to test a list specification document before running List Processing.

To test a list specification document:

1 Make sure the document you are testing is in the Current Document Block.

   If it is not, use the SEL option to select the document.

2 Enter T (Test Specification) at the List Processing Menu.

If there are no errors in the document, List Processing tells you with a screen message:

```
Specification file checks out all right
```

You can continue with List Processing or choose one of the other options from the List Processing Menu.

If the list specification document contains errors, the text that caused the first error is displayed with an error message.

To correct the error:

Edit your list specification document.

Words and Phrases in a List Specification: Summary

Figure 4-1 shows the words and phrases you use in a list specification document. In Figure 4-1, parentheses indicate optional items. The ellipses indicate that items may be repeated.

```
(IF field name=field value (range)

          (OR=field value (range),,,)

          (AND field name=field value (range)

                    (OR=field value (range),,,),,,)

          (BUT NOT IF field name=field value (range)

                    (OR=field value (range),,,),,,)

THEN) PROCESS RECORD

(OR,,,)

where range is one of these:

           THROUGH <field value>
           OR MORE
           OR LESS
```

Figure 4-1 Words and Phrases in a List Specification

NOTE: The field value before THROUGH, OR MORE, and OR LESS, and the field value after THROUGH, must be in angle brackets and the field tested must be numeric.

Table 4-1 summarizes the words and phrases used in a list specification document. Use this table once you are familiar with List Processing.

Table 4-1 Words and Phrases in a List Specification

Use	To
AND field name = field value	Select records with test values in more than one field.
BUT NOT IF field name = field value	Among records you are selecting, reject records with a test value in a field.
IF...THEN	Test if one or more fields in each list document record meets test conditions. IF...THEN PROCESS RECORD is an IF-statement.
OR	Select records that satisfy at least one of two IF-statements.
OR = field value	Select or reject records with at least one of two or more test values in certain fields.
OR LESS	Test a field for a value equal to or less than a test value. The field value before OR LESS must be in angle brackets.
OR MORE	Test a field for a value equal to or greater than a test value. The field value before OR MORE must be in angle brackets.
PROCESS RECORD	Process all records in a list document that meet the test conditions. If there is no IF...THEN, all records are processed.
THROUGH field value	Test a field for a value equal to or greater than the preceding test value and less than or equal to the following test value. The field values before and after THROUGH must be in angle brackets.

(continued)

Table 4-1 Words and Phrases in a List Specification (Cont.)

Use	To
<?>	Test for the presence of exactly one character, which can be any character, in a field value.
<*>	Test for any number of characters, which can be any characters at all, or no characters, in a field value.
<>, enclosing a field value	Test for a numeric value if the field name is numeric, or test for a sequence of digits if the field name is not numeric.

5

List Processing Math

This chapter explains how to use List Processing Math to:

- Perform addition, subtraction, multiplication, and division on numbers in list documents
- Format your results
- Check for errors in your formulas, format strings, and input data

With List Processing Math, you set up calculations to be performed on field values in list document records. List Processing Math performs the calculations and puts results in the output document.

If you change the field values and run List Processing Math again, List Processing Math calculates the new results.

List Processing Math and Editor Math

In many ways, List Processing Math is like Editor Math, which is documented in *WPS-PLUS Editor Functions*.

Table 5-1 summarizes the differences between Editor Math and List Processing Math.

Table 5-1 Editor Math and List Processing Math

Editor Math	List Processing Math
Starts calculations when you are using the editor and you move the cursor through a BEGIN control block.	Starts calculations during List Processing when a FORMULA control block is reached.
Gets information from decimal-aligned tab positions.	Gets information from input fields in the list document. Input fields are the fields identified on the left of equals signs in FORMULA commands.
Uses all possible formulas on every line of the table.	Uses all possible formulas on every record of the list until another control block is reached.
Answers appear in the table.	Answers appear in the output document where indicated by result fields in the form document. A result field is a field identified on the right of an equals sign in a FORMULA command.
Reworks calculations whenever you scroll through the table while in the WPS-PLUS Editor.	Reworks calculations whenever you run List Processing with the list document.

List Processing Math Commands

To use List Processing Math:

Put List Processing Math commands in control blocks at the top of the list document or between records.

Table 5-2 shows the five List Processing Math commands:

Table 5-2 List Processing Math Commands

Use	To
WPSMATH	Start List Processing Math. Without a FORMULA command in the same control block, the WPSMATH command stops List Processing Math.
FORMULA	Identify input fields, result fields, and the arithmetic operations you want List Processing Math to perform.
SET	Assign a value to a field.
TRUNCATE	Drop extra digits from a decimal fraction without rounding off. The number of digits in the fraction after truncation is the number specified in the format string, or at most six if there is no format string. Trailing zeros are dropped.
ROUND	Round off a decimal fraction. The fraction is rounded off to the number specified in the format string. Trailing zeros are dropped.

Starting List Processing Math

List Processing Math starts calculating when it comes to a FORMULA control block in the list document. A FORMULA control block contains a WPSMATH command and at least one FORMULA command.

For example:

```
------------------------- START CONTROL -------------------------
WPSMATH
FORMULA <:PRICE> * 0.1 = <:TAX> "$,.99"
FORMULA <:PRICE> + <:TAX> = <:TOTAL> "$,.99"
------------------------- END   CONTROL -------------------------
```

NOTE: For information about format strings, such as "$,.99," see the section Customized Format.

To perform calculations on all records in a list document:

Put FORMULA commands in a control block before the first list document record.

To perform calculations on certain records only:

1 Put FORMULA commands in a control block before the records to be used.

2 Put a null control block after the records.

See the section Stopping List Processing Math.

To change the calculations you are performing:

Insert another FORMULA control block where the new calculations start.

The earlier set of calculations stops at that point.

For the steps to follow in creating a FORMULA control block, see Chapter 1.

Stopping List Processing Math

List Processing Math stops calculating when it encounters:

- The end of the list document
- A null control block

A null control block contains a WPSMATH command but no FORMULA commands.

A typical null control block looks like this:

```
------------------------ START CONTROL ------------------------
WPSMATH
------------------------ END   CONTROL ------------------------
```

To create a null control block:

1 Press Gold CMND to insert a START CONTROL marker.

2 Type WPSMATH.

3 Press RETURN.

4 Press Gold CMND to insert an END CONTROL marker.

List Processing Math starts calculating again if it comes to a FORMULA control block.

HINT. You can include SET commands in a null control block to set fields or store values. For more information about the SET command, see WPS-PLUS Editor Functions.

FORMULA Command Guidelines

FORMULA commands perform all of your calculations.

When you write FORMULA commands, follow these guidelines:

1 Only use numeric fields in FORMULA commands. Begin each field name in a FORMULA command with a left angle bracket and colon, <:.

2 Put input fields to the left of the equals sign, =, in FORMULA commands.

3 Use the symbols, + - * and /, to tell List Processing Math which arithmetic operations to perform.

4 Put result fields to the right of the equals sign.

If a result field is empty, List Processing Math assigns a value of zero.

5 Put as many FORMULA commands as you need in a control block.

Allowable Characters

Besides field names, the only characters allowed in List Processing Math formulas are those shown in Table 5-3.

Table 5-3 Allowable Characters

Character	Description
0 1 2 3 4 5 6 7 8 9	Numerals, used to write constants. Constants are numbers you write into formulas. They do not change from one line of a table to the next, as field values usually do.
. ,	Decimal point and comma, which can be used to write constants.
$	A dollar sign, indicating cash value.
+	A plus sign, indicating addition or positive numbers.
-	A minus sign, indicating subtraction or a negative number.
*	An asterisk, indicating multiplication.
/	A slash, indicating division.
( )	Parentheses, used to group operations together. The parentheses must always match. For every right parenthesis,), there must be exactly one left parenthesis, (.
=	An equals sign, indicating the end of the calculation. Every formula must contain only one equals sign. Do not put + - * / (or) on the right side of an equals sign. Instead, you must follow the equals sign with the name of a field where you want the answer to go.
	A space. You can put spaces in a FORMULA command. Editor Math ignores them, but they can make formulas easier to read. Do not put extra spaces in a field name, or Editor Math will not recognize the name.

Constants

A constant is any number in a formula. For example:

```
FORMULA <:PRICE> * 0.1 = <:TAX>
```

In this example, 0.1 is a constant.

You can add or subtract a constant, or multiply or divide by a constant. A constant can be negative or positive, a whole number, or a fraction.

Processing List Processing Math

When you run List Processing, List Processing Math:

1. Does calculations in the order in which the formulas appear in a FORMULA control block.
2. Applies every formula in the control block to the records that follow.
3. Gives you an error message if there is an error in your control block.

   Press RETURN to edit the command.

List Processing Math: Complete Example

Figure 5-1 shows a list document with List Processing Math commands, a sample form document, and a sample output document.

The first FORMULA command multiplies the total sales for each sales representative by 0.1 to produce the commission.

The second FORMULA command subtracts the commission and expenses for each representative from the sales to produce the profit.

The form document indicates where the results go in the output document. The result fields are <:COMMISSION> and <:PROFIT>.

The sample output shown is part of a sales report. The sales report contains field values from the list document and results calculated by List Processing Math.

LIST DOCUMENT:

```
This is a list of our sales force, their sales for the month, and their
expenses. Calculated data will include sales commission and profit.

--------------------------- START CONTROL ----------------------------
WPSMATH
FORMULA <:SALES> * 0.1 = <:COMMISSION> "$99,999.99"
FORMULA <:SALES> - <:COMMISSION> - <:EXPENSES> = <:PROFIT> "$99,999.99"
--------------------------- END   CONTROL ----------------------------
<NAME>Joan Hamett
<AREA>Boston
<:SALES>$8809.00
<:EXPENSES>$550.78
<>

<NAME>Sara Hayes
<AREA>New York
<:SALES>$14,238.67
<:EXPENSES>$924.62
<>
```

FORM DOCUMENT:

```
Sales Representative Report

<!S><NAME>, working in <AREA>, made <:SALES>.  Expenses were
<:EXPENSES> and the commission was <:COMMISSION>, leaving a profit of
<:PROFIT>.

<!E>
```

OUTPUT DOCUMENT:

```
Sales Representative Report

Joan Hamett, working in Boston, made $8809.00.  Expenses were
$550.78 and the commission was $880.90, leaving a profit of $7,377.32.

Sara Hayes, working in New York, made $14,238.67.  Expenses were
$924.62 and the commission was $1,423.87, leaving a profit of $11,890.18.
```

Figure 5-1 List Processing Math Commands

Totals and Subtotals

To calculate totals and subtotals:

Use FORMULA commands with accumulator fields.

Accumulator fields store running totals.

For example, suppose you need to total the cost of your inventory. Here is a sample list document:

```
-------------------------- START CONTROL -------------------------
WPSMATH
FORMULA <:COST> * <:NUMBER IN STOCK> = <:INVENTORY COST> "$,,99"
FORMULA <:TOTAL COST> + <:INVENTORY COST> = <:TOTAL COST> "$,,99"
-------------------------- END   CONTROL -------------------------
<:PART NUMBER>01
<PART NAME>Binder
<:COST>$1.19
<:NUMBER IN STOCK>47
<>

<:PART NUMBER>02
<PART NAME>Cellophane tape
<:COST>$1.59
<:NUMBER IN STOCK>97
<>
```

The first command in the control block calculates the in-stock value of an inventory item.

The second command adds the in-stock value of each inventory item to the running total in the accumulator field, <:TOTAL COST>.

NOTE: If you have used an accumulator before in a list document, use a SET command to set the accumulator to zero before starting a total or subtotal. For example: SET 0 = <:TOTAL SALES REPS>.

To display a total in your output document:

Include the accumulator in your form document, in the trailer, for example.

The next example shows a way to calculate subtotals. Suppose sales representatives are sorted into North and South regions in your list document. You want to subtotal the sales for each region and keep a

running total of all sales too. Your list document contains sales representatives in the North region, followed by sales representatives in the South region.

Figure 5-2 shows the list document with List Processing Math commands:

```
SALES COMMISSIONS REPORT:

THIS LIST DOCUMENT CONTAINS RECORDS OF SALES COMMISSIONS IN BOTH
REGIONS, NORTH AND SOUTH.
-------------------------- START CONTROL -------------------------
WPSMATH
FORMULA <:NORTH> + <:COMMISSION> = <:NORTH> "$*,99999.99"
FORMULA <:GRAND TOTAL> + <:COMMISSION> = <:GRAND TOTAL> "*,99999.99"
-------------------------- END   CONTROL -------------------------

NORTH DISTRICT:

<LAST>Tucker
<FIRST>Fred
<:EMPLOYEE NUMBER>001
<:COMMISSION>$13,880.25
<>
<LAST>Bascomb
<FIRST>Roger
<:EMPLOYEE NUMBER>002
<:COMMISSION>$13,009.00
<>
<LAST>Robertson
<FIRST>Michael
<:EMPLOYEE NUMBER>003
<:COMMISSION>$10,400.00
<>
-------------------------- START CONTROL -------------------------
WPSMATH
FORMULA <:SOUTH> + <:COMMISSION> = <:SOUTH> "$*,99999.99"
FORMULA <:GRAND TOTAL> + <:COMMISSION> = <:GRAND TOTAL> "$*,99999.99"
-------------------------- END   CONTROL -------------------------

SOUTH DISTRICT:

<LAST>Whitcomb
<FIRST>David
<:EMPLOYEE NUMBER>004
<:COMMISSION>$14,000.90
<>
<LAST>Stuart
<FIRST>Roger
<:EMPLOYEE NUMBER>005
<:COMMISSION>$15,555.00
<>
```

Figure 5-2 Subtotals

In Figure 5-2, the command that calculates the subtotal for the North region is:

```
FORMULA <:NORTH> + <:COMMISSION> = <:NORTH> "$*,99999.99"
```

The command that calculates the total commissions is:

```
FORMULA <:GRAND TOTAL> + <:COMMISSION> = <:GRAND TOTAL> "$*,99999.99"
```

A second WPSMATH control block starts subtotalling the South region.

The command that calculates the subtotal for the South region is:

```
FORMULA <:SOUTH> + <:COMMISSION> = <:SOUTH> "$*,99999.99"
```

When List Processing encounters the second control block, the first set of calculations stops. The second control block restarts the calculation for the grand total by repeating a command from the first control block.

Here is the form document.

```
COMMISSIONS EARNED IN ALL DISTRICTS (Q4)

<!S>
Employee <:EMPLOYEE NUMBER>: <:COMMISSION><!E>


SUMMARY:

        NORTH District employees: <:NORTH>
        SOUTH District employees: <:SOUTH>

        TOTAL commissions: <:GRAND TOTAL>
```

The trailer in the form document contains the North and South subtotals and the grand total.

Here is the output document:

```
COMMISSIONS EARNED IN ALL DISTRICTS  (Q4)

Employee 001: $13,880.25
Employee 002: $13,009.00
Employee 003: $10,400.00
Employee 004: $14,000.90
Employee 005: $15,555.00


SUMMARY:

        NORTH District employees: $37,289.25
        SOUTH District employees: $29,555.90

        TOTAL commissions earned: $66,845.15
```

Averages

When calculating averages, you can use the special accumulator <:LINES> to total the number of records processed.

To total the number of records processed:

> Put the accumulator <:LINES> in the FORMULA control block at the top of the list document.

<:LINES> counts the number of list document records processed. Divide a total by the value in <:LINES> to produce an average over the records processed.

For example, suppose you need to find the average value of an inventory item. These commands calculate the totals you need to find the average:

```
------------------------- START CONTROL -------------------------
WPSMATH
FORMULA <:LINES> + 1 = <:LINES>
FORMULA <:TOTAL> + <:COST> = <:TOTAL> "$,.99"
-------------------------- END   CONTROL -------------------------
```

The first formula increases the value of <:LINES> by one with each record processed. Initially, the value for <:LINES> is zero.

The second formula calculates the total cost of your inventory items by adding the cost in each record to <:TOTAL>.

To find the average cost of an inventory item, divide <:TOTAL> by <:LINES>:

```
FORMULA <:TOTAL> / <:LINES> = <:AVERAGE> "$.99"
```

To see the values in <:TOTAL>, <:LINES>, or <:AVERAGE>, put the field name in the trailer of the form document.

Here is the list document with commands included:

```
-------------------------- START CONTROL ----------------------
WPSMATH
FORMULA <:LINES> + 1 = <:LINES>
FORMULA <:TOTAL> + <:COST> = <:TOTAL> "$.99"
FORMULA <:TOTAL> / <:LINES> = <:AVERAGE> "$.99"
-------------------------- END   CONTROL ----------------------
<:PART NUMBER>01
<PART NAME>Binder
<:COST>$1.19
<:NUMBER IN STOCK>47
<>

<:PART NUMBER>02
<PART NAME>Cellophane tape
<:COST>$1.59
<:NUMBER IN STOCK>97
<>
```

Here is a form document with a trailer that shows the average cost of inventory items.

```
INVENTORY REPORT: 11-5-84

<!S>
<:PART NUMBER> <PART NAME>

       <:NUMBER IN STOCK> in stock @ <:COST>

<!E>

Inventory includes <:LINES> parts, at an average cost of <:AVERAGE>
each.
```

Here is a sample output document:

```
INVENTORY REPORT: 11-5-84

01 Binder

     47 in stock @ $1.19

02 Cellophane tape

     97 in stock @ $1.59

Inventory includes 2 parts, at an average cost of $1.39 each.
```

Default Format

Unless you include a format string in a FORMULA command, List Processing Math writes results in default format. These are the characteristics of default format:

- Dollar signs and commas are not included.
- If a number is negative, a minus sign is in front of the number.
- If the number is not a whole number, the number contains a decimal point and digits for the fractional part.
- No zeros are displayed after the last nonzero digit of a decimal fraction.
- Numbers can have a maximum of 13 digits. Up to six of those digits can follow the decimal point.
- Fractions of more than six digits are rounded off.

Use format strings when you want results in a customized format, such as with dollar signs ($), or rounded off to a certain number of decimal places.

Customized Format

To write results in a different format:

> Put a format string in quotation marks at the end of your FORMULA command.

The characters you can include in a format string are - ( ) $ * , . and 9.

For example, the format string in the following command tells List Processing Math to round off its answer to one decimal digit:

```
FORMULA <:D2> + <:T2> = <:T2> ",9"
```

Decimal Fractions

To format results containing decimal fractions:

> Decide on the largest number of digits you need right of the decimal point, then use that number of 9s after the decimal point in your format string.

You can drop some decimal digits by including fewer 9's after the decimal point than the full answer requires. You can include at most six 9s to the right of the decimal point.

For example:

```
FORMULA   <:COST> * <:ITEMS> = <:TOTAL>   ",999"
```

The format string in this example tells List Processing Math to round off the result to three decimal places.

In default format, decimal fractions are written with one zero to the left of the decimal point. As a result, one half is written as 0.5. You cannot avoid displaying the initial zero.

To drop all decimal digits and round answers to the nearest integer:

> Use the format "9."

You do not need to include 9s left of the decimal point unless your format string includes an asterisk. See the section Asterisks.

Commas

If your results are large numbers, use commas to group digits for easier reading. In a format string, a comma must come after any dollar signs and before a decimal point. One comma is all you need.

The following strings give the same results:

```
"$9,999,999.99"     "$,,99"
```

Dollar Signs

If you include the dollar sign ($) at the beginning of your format string, List Processing puts a dollar sign in front of the answer.

To format results in dollars and cents:

Use the format string "$,.99"

Trailing Minus Sign

A trailing minus sign in a format string tells List Processing Math to put the minus sign after a negative number (775.01-) instead of before it. In a format string, a trailing minus sign must be the last symbol before the closing quotation mark. Examples of the trailing minus format are:

```
"*9999.99-"          "*999,999-"              "$,,99-"
```

Parentheses

Parentheses in a format string tell List Processing Math to write negative numbers in parentheses instead of with a minus sign. Here is a negative result written with parentheses:

```
(999.01)
```

Make parentheses the first and last symbols in a format string:

```
"($999.99)"          "( )"                    "(*,999999)"
```

Notice that you do not have to include 9s to use parentheses format. If you include just "()" as your format string, List Processing Math writes its answers in default format, except that it uses parentheses instead of a leading minus sign to indicate negative numbers.

Asterisks

You can fill out results with asterisks. That way, you can keep a column of results even on the right. For example:

```
$84,000.00
$*4,343.00
$***778.00
```

To fill out a result with asterisks:

1. Put one asterisk after any dollar sign and before any commas or 9s.
2. Put in as many 9s left of the decimal point as there are digits and commas in the largest result you expect.

List Processing writes results so the number of asterisks, commas, and numerals left of the decimal point is the same as the number of 9's in the format string.

You can put a maximum of 13 9's in a format string, including a maximum of six right of the decimal point.

For example:

```
FORMULA <:COST> * <:ITEMS> = <:TOTAL>   "$*999,999,999.99"
FORMULA <:COST> * <:ITEMS> = <:TOTAL>   "$*,999999999.99"
```

The format strings in these commands produce the same results. If a result contains fewer than nine digits and commas left of the decimal point, List Processing Math adds asterisks until the total of digits, commas, and asterisks is nine. For example:

```
$***94,000.00
```

If a result contains nine or more digits and commas left of the decimal point, no asterisks are added. For example:

```
$4,320,213.00
```

Truncation and Rounding

By default List Processing Math rounds off the decimal digit corresponding to the last 9 in a format string.

To truncate results instead of rounding off:

Use the TRUNCATE command.

Truncation simply drops extra digits. If List Processing Math truncates 0.796, it writes out 0.79.

Type the TRUNCATE command in a control block on a line by itself on any line after the word WPSMATH. For example:

```
---------------------- START CONTROL ----------------------
WPSMATH
FORMULA   <:COST> / 10 = <:REDUCTION> "$,9.99"
TRUNCATE
----------------------- END   CONTROL ----------------------
```

To resume rounding off:

Use a ROUND command.

Type ROUND on a line by itself in a control block.

Always give a separate line to a TRUNCATE or ROUND command.

Records Selected for Processing

List Processing Math works only on the selected records. If a list specification, or a setting on the List Processing Documents Selected Form, excludes a record, List Processing Math does not use field values in the record for calculations.

List Processing Math Error Messages

When List Processing Math finds an error, processing stops and an error message displays. Part of the line containing the error is displayed above the message. The line ends at the point where List Processing Math could not understand the command or the number. The location of the error is marked with a caret, ^.

For example:

```
FORMULA <:CREDIT> + <:TOTAL> <:TOTAL>
-------------------------------------^
Formula error
```

In this example, an equals sign was omitted from a FORMULA command. Notice that the error is not always at the point where List Processing Math ends the line. Sometimes, you can correct the error by changing something earlier in the line.

Most errors occur because List Processing Math cannot understand what your commands mean or what numbers you want to use. Use the following checklists to avoid and correct errors.

FORMULA Command Checklist

To avoid making formula notation errors, or to correct them, make sure that:

- The characters + - * / and = are always followed by field names, constants, or text in parentheses, ( ).
- All field names, constants, and pieces of text in parentheses, ( ), are separated from their neighbors by + - * / or =.
- There is exactly one right parenthesis,), for every left parenthesis, (.
- There is a RETURN at the end of every formula, or, if there is a format word, at the end of the format word.
- There is exactly one equals sign, =, in every FORMULA and SET command, with exactly one field name after it.
- The commands WPSMATH, TRUNCATE, and ROUND each have a line to themselves.
- Every field name starts with <: and ends with >.
- The only characters in constants are 0 1 2 3 4 5 6 7 8 9 , + - . and $.
- The characters + - . $ appear no more than once in any constant, in the correct places.
- No constant has more than 13 digits, or more than 6 digits to the right of the decimal point.
- The command does not end too soon. Make sure there is an equals sign, =, a constant, or a complete field name to the left of the equals sign, and a complete field name to the right of the equals sign.

Format Checklist

To avoid making errors and to correct errors in format strings, make sure that:

- Each format string is enclosed in quotation marks.
- The only characters in format strings are - ( ) * $, . 9.
- The characters - ( ) $ * . appear no more than once.
- The minus sign, -, is either first or last, if it appears.
- Parentheses, ( ), and the minus sign, -, do not appear in the same format string.
- There is exactly one right parenthesis,), for every left parenthesis, (.
- The dollar sign, $, comes before everything except the minus sign, -, or left parenthesis, (.
- The asterisk, *, comes before 9 , .) or a trailing minus sign, -.
- There are no commas after the decimal point.
- There are no more than thirteen 9s in the format string and no more than six 9s to the right of the decimal point.
- Format strings containing the characters * - . always contain 9s.

Input Field Checklist

To avoid making errors and to correct errors involving numbers in input fields, make sure that:

- There are no empty pairs of parentheses, ( ).
- The only characters in the field value are 0 1 2 3 4 5 6 7 8 9 $ * , . ( ).
- The characters $. + - ( ) appear no more than once.
- The characters $ * + (appear before any digits.
- The plus sign, +, comes first, if it appears.
- The minus sign, -, is either first or last, if it appears.

- The dollar sign, $, comes before any asterisk, *, and after the plus sign, +, or the minus sign, -.
- There is exactly one right parenthesis,), for every left parenthesis, (.
- Parentheses come first and last, if they appear.
- Parentheses and the minus sign, -, do not appear in the same field value.
- There are no more than 13 digits in the field value, and no more than six digits to the right of the decimal point.

6

Sorting a List Document

This chapter explains how to create sort specification documents to sort on more than one field. In addition, the chapter contains:

- Sample output from Sort Processing
- The Sort Alphabet (which gives the order used in sorting)

Sort Specification Documents

A sort specification document tells List Processing how to sort records in a list document.

You can sort records by one or more fields. For example, you can sort a mailing list by state so records for Alabama come first, followed by all records for Alaska, Arizona, Arkansas, and so on. You can sort a mailing list by state and city so records for Allentown, Pennsylvania are grouped together, followed by records for Altoona, Pennsylvania, followed by those for Bethlehem, Pennsylvania and so on.

You can sort records in ASCENDING alphabetic/numeric order or in DESCENDING alphabetic/numeric order.

For the steps to follow in creating a sort specification document to sort on one field, see Chapter 1.

Sorting on More Than One Field

You can use as many fields as you need to use in sorting.

Here is a sample sort specification:

```
SORT ON <STATE> IN ASCENDING ORDER AND

SORT ON <CITY> IN ASCENDING ORDER AND

SORT ON <STREET> IN ASCENDING ORDER
```

This sort specification sorts list document records by the fields <STATE>, <CITY>, and <STREET>. Sort Processing organizes records by states, and within states, by cities, and within cities, by street addresses.

To sort on more than one field:

1. Put the primary sort field on the first line of the sort specification document.

   The primary field is usually the largest classification among the fields sorted.

2. Specify the sort order for the primary field, ASCENDING or DESCENDING.

3. End the line with AND.

   AND tells Sort Processing that another line follows.

4. Put other sort fields on subsequent lines of the sort specification document, arranged from largest to smallest classification, and specify their sort orders. End each line except the last line with AND.

In the previous example, the largest classification is <STATE>, so <STATE> is the primary sort field, the field used first in sorting records. The next largest classification is <CITY>, so <CITY> is the secondary sort field. The smallest classification is <STREET>, so <STREET> is the last sort field.

Figure 6-1 shows sample output:

```
<LAST>Baines
<FIRST>Louis
<STREET#>432
<STREET>Drayton Avenue
<CITY>Anaheim
<STATE>CA
<ZIP>92803
<>

<LAST>Whipple
<FIRST>Roger
<STREET#>74
<STREET>Main Street
<CITY>Campbell
<STATE>CA
<ZIP>95008
<>

<LAST>Hardy
<FIRST>Ted
<STREET#>900
<STREET>Thomason Avenue
<CITY>Campbell
<STATE>CA
<ZIP>95008
<>

<LAST>Billingham
<FIRST>Matthew
<STREET#>453
<STREET>Orchard Road
<CITY>Londonderry
<STATE>NH
<ZIP>03053
<>
```

Figure 6-1 List Sorted on <STATE>, <CITY>, and <STREET> Ascending Order

In Figure 6-1, California records occur before New Hampshire records. Within California records, Anaheim records occur before Campbell records. Within Campbell records, Main Street records occur before Thomason Avenue records.

Examples of Sort Output Documents

The next example shows a sort in ascending order on two fields, <STATE> and <LAST>. Since <STATE> is the primary field, Sort Processing sorts by the <STATE> field first. Then Sort Processing sorts by the <LAST> field. The sort specification document is:

```
SORT ON <STATE> IN ASCENDING ORDER AND

SORT ON <LAST> IN ASCENDING ORDER
```

Figure 6-2 shows the output document.

```
<LAST>Hutchinson
<FIRST>William
<ADD>3708 Loam Avenue
<CITY>Anaheim
<STATE>CA
<ZIP>92803
<>

<LAST>Robertson
<FIRST>William
<ADD>910 Hedge Road
<CITY>Campbell
<STATE>CA
<ZIP>95008
<>

<LAST>Briand
<FIRST>Robert
<ADD>159 Colby Street
<CITY>Nashua
<STATE>NH
<ZIP>03063
<>

<LAST>Franklin
<FIRST>Samuel
<ADD>608 Knoll Road
<CITY>Manchester
<STATE>NH
<ZIP>03101
<>

<LAST>Wilson
<FIRST>Sarah
<ADD>175 Inwood Avenue
<CITY>Manchester
<STATE>NH
<ZIP>03101
<>
```

Figure 6-2 List Sorted on <STATE> and <LAST> in Ascending Order

The next example shows a sort on the field name <ITEM#>. Some field values in the <ITEM#> field start with letters, others with numbers, still others with symbols or spaces. (Refer to Table 6-1 to see how sort places special symbols and numbers in ascending order.) The sort specification document is:

```
SORT ON <ITEM#> IN ASCENDING ORDER
```

Figure 6-3 shows the sort output document.

```
<ITEM#>0008
<DESCRIP>DECmat 4'x6'
<SHIPPING WT>10 lbs.
<:UNIT PRICE>125.00
<>

<ITEM#>0074
<DESCRIP>Mylar Ribbon
<SHIPPING WT>1 lb.
<:UNIT PRICE>42.00
<>

<ITEM#>3009
<DESCRIP>WP rulers - pkg. of 10
<SHIPPING WT>.05 lbs
<:UNIT PRICE>$20.00
<>

<ITEM#>8976
<DESCRIP>Serial Interface Switch
<SHIPPING WT>2 lbs.
<:UNIT PRICE>$235.00
<>

<ITEM#>A0076
<DESCRIP>Acoustic cover
<SHIPPING WT>40 lbs.
<:UNIT PRICE>425.00
<>

<ITEM#>T0074
<DESCRIP>Fanfold Paper
<SHIPPING WT>27 lbs.
<:UNIT PRICE>43.00
<>
```

Figure 6-3 List Sorted on <ITEM#> in Ascending Order

The next example shows a sort on two fields. The primary field is <PAYMT> and the sort is descending. The secondary field is <LAST> and the sort is ascending.

The sort specification document is:

```
SORT ON <PAYMT> IN DESCENDING ORDER AND
SORT ON <LAST> IN ASCENDING ORDER
```

Figure 6-4 shows the sort output document.

```
<LAST>Emery
<FIRST>Charles
<PAYMT>credit
<>

<LAST>Maclean
<FIRST>Roxanne
<PAYMT>COD
<>

<LAST>Robertson
<FIRST>William
<PAYMT>COD
<>

<LAST>Brindel
<FIRST>Robert
<PAYMT>cash
<>

<LAST>Kadis
<FIRST>Freeman
<PAYMT>cash
<>

<LAST>Spell
<FIRST>Sarah
<PAYMT> credit
<>
```

Figure 6-4 List Sorted on <PAYMT> Descending and <LAST> Ascending

Notice that the last record contains a field value that begins with an accidental space. The space causes the record to be sorted last. Spaces are first in the Sort Alphabet. They are placed last when the sort is DESCENDING. Misplaced tabs can also affect your sorting.

The next example shows a sort on a numeric field. The list document is arranged in descending order according to the numeric value of the <:UNIT PRICE> field.

The sort specification document is:

```
SORT ON <:UNIT PRICE> IN DESCENDING ORDER
```

Figure 6-5 shows the sort output document.

```
<ITEM#>A0076
<DESCRIP>Acoustic cover
<SHIPPING WT>40 lbs.
<:UNIT PRICE>425.00
<COD>Y
<>

<ITEM#>8976
<DESCRIP>Serial Interface Switch
<SHIPPING WT>2 lbs.
<:UNIT PRICE>$235.00
<COD>Y
<>

<ITEM#>/008
<DESCRIP>DECmat 4'x6'
<SHIPPING WT>10 lbs.
<:UNIT PRICE>125.00
<CREDIT>Y
<>

<ITEM#>T0074
<DESCRIP>Fanfold Paper
<SHIPPING WT>27 lbs.
<:UNIT PRICE>43.00
<CREDIT>Y
<>

<ITEM#>0074
<DESCRIP>Mylar Ribbon
<SHIPPING WT>1 lb.
<:UNIT PRICE>42.00
<CASH>Y
<>

<ITEM#>3009
<DESCRIP>WP rulers - pkg. of 10
<SHIPPING WT>.05 lbs
<:UNIT PRICE>$20.00
<CASH>Y
<>
```

Figure 6-5 List Sorted on <:UNIT PRICE> in Descending Order

Sort Alphabet

Table 6-1 shows the Sort Alphabet, which is the sequence of symbols, numbers, and letters used by Sort Processing.

Table 6-1 Sort Alphabet

Sort Value	Character	Sort Value	Character	Sort Value	Character
1	space	23	6	45	L l
2	!	24	7	46	M m
3	"	25	8	47	N n
4	#	26	9	48	O o
5	$	27	:	49	P p
6	%	28	;	50	Q q
7	&	29	<	51	R r
8	'	30	=	52	S s
9	(	31	>	53	T t
10	)	32	?	54	U u
11	*	33	@	55	V v
12	+	34	A a	56	W w
13	,	35	B b	57	X x
14	-	36	C c	58	Y y
15	.	37	D d	59	Z z
16	/	38	E e	60	[{
17	0	39	F f	61	\ |
18	1	40	G g	62	] }
19	2	41	H h	63	^ ~
20	3	42	I i	64	_
21	4	43	J j	65	
22	5	44	K k		

NOTE: DEC Multinational Characters are sorted with other characters. All a's are grouped with other a's, b's with b's, and so on. In foreign languages, the sort order is often different from the order used for English. Technical characters are sorted after standard characters in ascending order, and before standard characters in descending order. See WPS-PLUS Editor Functions for more information about multinational and technical characters.

Sorting Missing Fields or Field Values

When you sort a list document by a field that is missing in some records, or by a field that is missing a field value in some records, the records with the missing field or field value are not sorted correctly. If the sort order is DESCENDING (reverse alphabetical order), these records are placed first. If the sort order is ASCENDING (in alphabetical order), they are placed last.

Words and Phrases in a Sort Specification: Summary

Figure 6-6 shows the words and phrases used in a sort specification document.

```
sort specification (AND

sort specification,,,)

where sort specification is one of the following:

SORT ON field name in ASCENDING ORDER
SORT ON field name in DESCENDING ORDER
```

Figure 6-6 Words and Phrases in a Sort Specification

Table 6-2 summarizes the words and phrases used in a sort specification document.

Table 6-2 Words and Phrases in a Sort Specification

Use	To
SORT ON field name IN ASCENDING ORDER	Sort list documents in ascending alphabetic or numeric order by the value in the specified record field.
SORT ON field name IN DESCENDING ORDER	Sort list document records in descending alphabetic or numeric order by the value in the specified record field.
AND sort specification	Use more than one sort spec. The later specification is used for sorting after the earlier one is used.

7

Running List and Sort Processing

This chapter explains how to run List and Sort Processing. You run List and Sort Processing from the List Processing Menu.

To display the List Processing Menu:

> Enter LP (List Processing) from the Word and Document Processing Menu.

List Processing Menu

Figure 7-1 shows the List Processing Menu.

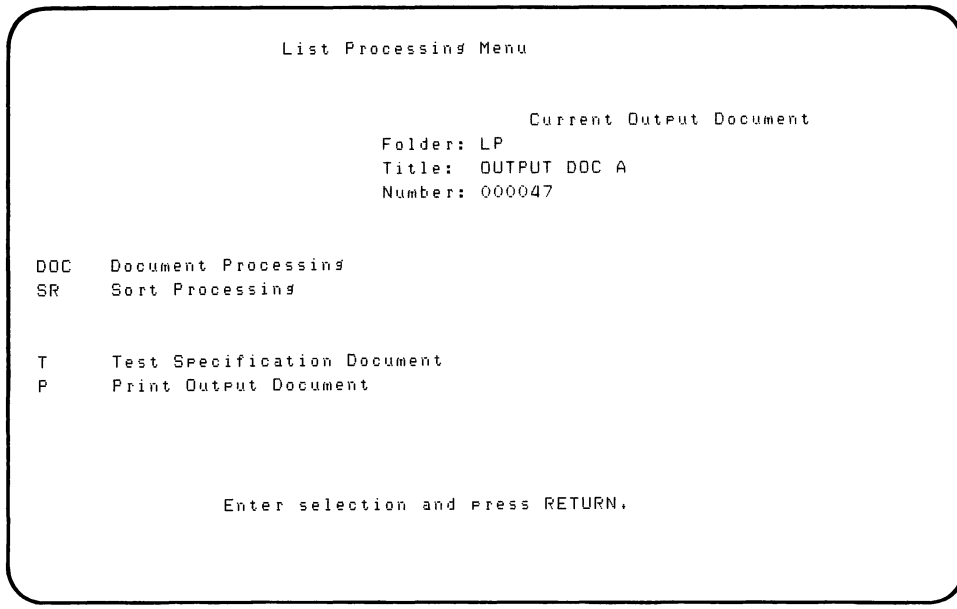


Figure 7-1 List Processing Menu

Table 7-1 describes the options on the List Processing Menu.

Table 7-1 List Processing Menu Options

Use		To
DOC	Document Processing	Select List Processing.
SR	Sort Processing	Select Sort Processing.
T	Test Specification Document	Test a list specification document without running List Processing.
P	Print Output Document	Print the output document identified in the Current Output Document Block. The Current Output Document Block is the window on the upper right of the List Processing Menu.

List Processing

To use List Processing:

1 Create the list, form, and list specification documents discussed in Chapters 2, 3, and 4.

   If your list specification document is long or complex, use the Test Specification Document (T) option to test the document. See Chapter 4.

2 Enter DOC (Document Processing) from the List Processing Menu to process the three documents and produce an output document.

When you select the Document Processing option, these forms are displayed:

- Selecting List Document Form
- Selecting List Specification Document Form
- Selecting Form Document Form

NOTE: After you run List Processing for the first time, the List Processing Documents Selected Form displays when you enter Document Processing. Make sure the documents identified on the form are the ones you want to use. To add or change a document, press Gold A to display the document selection forms. When you change information on a form, use RUB WORD OUT if you need to erase a number that appears in the number field.

Completing the Document Selection Forms

Each document selection form has a field for folder name, document title, and document number.

Figure 7-2 shows the Selecting List Document Form.

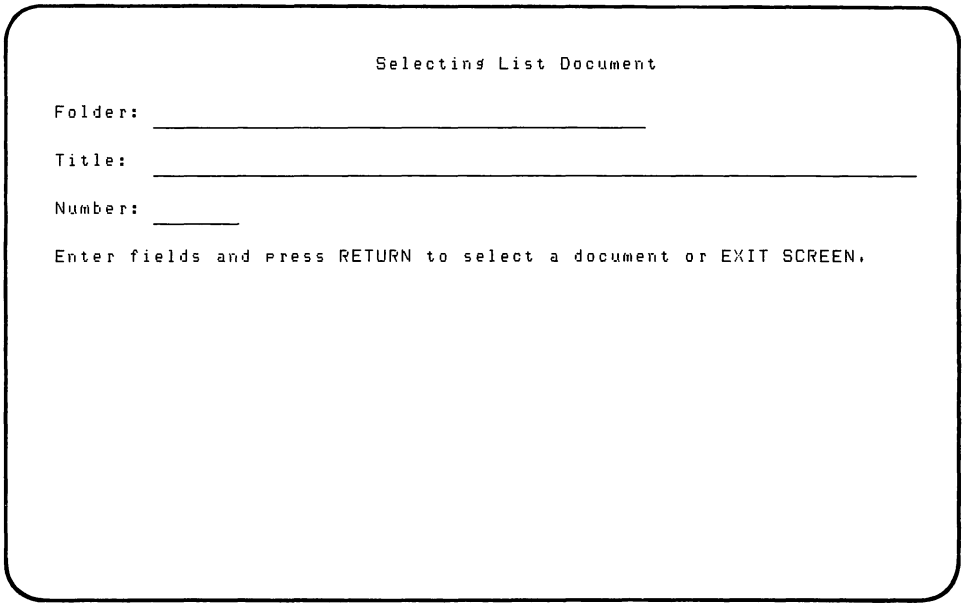


Figure 7-2 Selecting List Document Form

To complete a form:

1 Type the folder name and title of the document you want to process.

   Use the TAB and HYPH PUSH keys to move from field to field.

2 Press RETURN to display the next form.

NOTE: When selecting a document for List or Sort Processing, use the document number to distinguish between documents with the same name in the same folder. You can use Gold L from the number field to get the document number.

When you have completed these forms, the Select Output Document Option Menu displays. Table 7-2 describes the menu options.

Table 7-2 Select Output Document Option Menu

Use	To
O	Overwrite the current output document. The document you identify receives the output. Any text already in the document is destroyed. If no records are processed, you are left with an empty output document.
B	Add text to the bottom of the current output document. The document you identify receives the output. Any text already in the document remains, above the new output.
T	Add text to the top of the current output document. The document you identify receives the output. Any text already in the document remains, below the new output.
N	Create a new output document. The new document you identify receives the output.

If you enter O, B, or T, WPS-PLUS displays the Selecting Output Document Form, shown in Figure 7-3.

If you enter N, WPS-PLUS displays the Creating New Output Document Form, shown in Figure 7-4.

To complete either form:

1 Type the folder name and title of the output document.

2 Press RETURN.

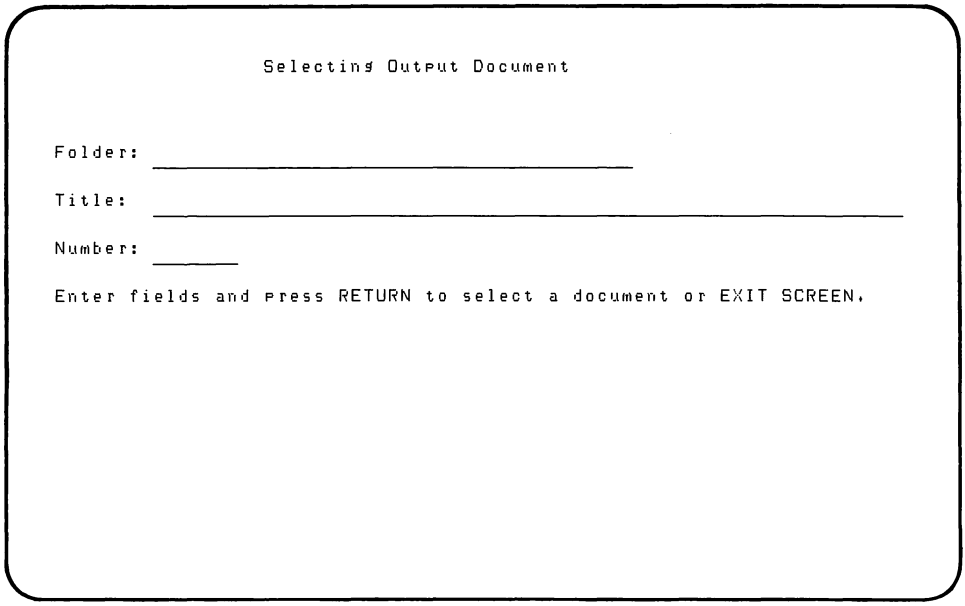


Figure 7-3 Selecting Output Document Form

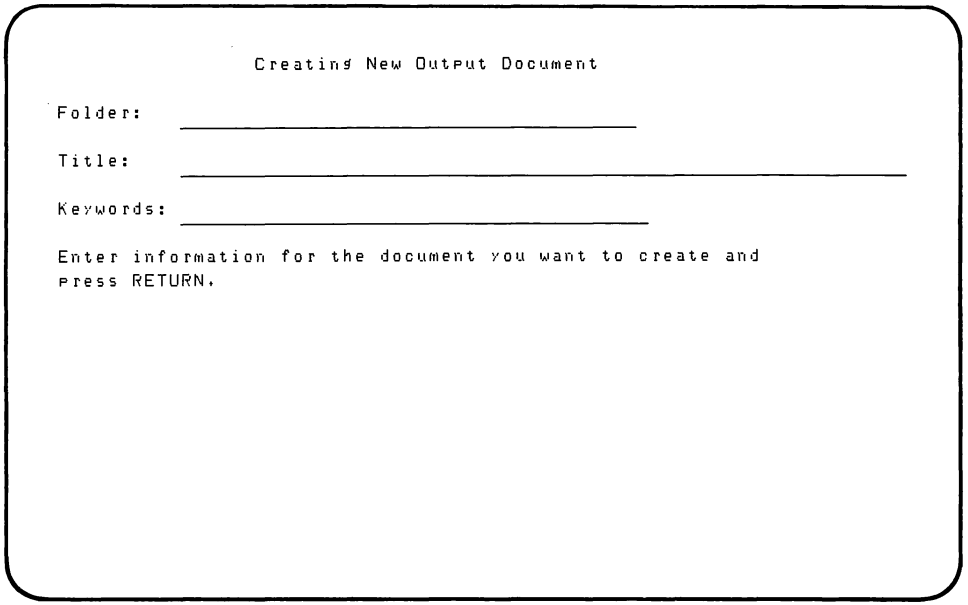


Figure 7-4 Creating New Output Document Form

The List Processing Documents Selected Form displays. Figure 7-5 shows the List Processing Documents Selected Form.

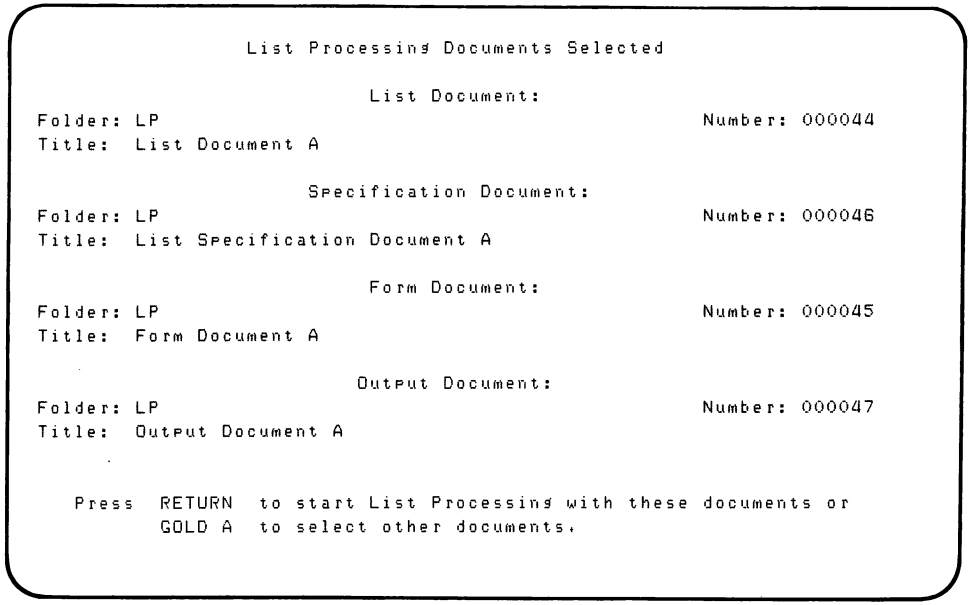


Figure 7-5 List Processing Documents Selected Form

How to Start List Processing

To start List Processing:

Press RETURN at the List Processing Documents Selected Form.

NOTE: You cannot use your terminal for other tasks while List Processing is running.

Start and Finish Settings

At the bottom of the List Processing Documents Selected Form are two settings that control the first and last records processed:

```
Start processing from this record
Process up to and including
```

The default settings are:

```
Start processing from this record: 1
Process up to and including:       0
```

To process all records in the list that meet the list specification requirements:

Use the default values for these settings.

Since record 0 is never reached, the default settings process every record that matches the list specification.

To process fewer records:

Change one or both of the settings.

For example, if 15 and 22 are your settings, only the 15th through the 22nd records of the list document are submitted to List Processing. For List Processing to use a record, the record must be in the range from 15 through 22 and its field values must match those in the selection specification.

WPS-PLUS resets the two settings to their defaults when List Processing is finished.

When you press RETURN to start List Processing, the two settings at the bottom of your screen are replaced by a window displaying:

- The number of records selected
- The number of records processed
- The number of records in the list document

You can use this information to locate errors. For example, if you are processing all records in your list document, but the number of records selected is zero, there is probably an error in your specification document.

When List Processing is completed, WPS-PLUS displays the List Processing Menu.

To stop List Processing at any time:

Press CTRL/C.

Error Messages

If List Processing finds an error, the text that caused the error is displayed and the error is marked with the symbol ^. For example:

```
IF <OVERDUE>=<2> OR MORE
THEN PROCESS RECA
-----------------------^
ERROR-- specification not understood starting at this point.
```

When you receive an error message, note which document caused the error, then edit the document.

Printing an Output Document

Use the Printing Output Document (P) option on the List Processing Menu to print an output document. The Print option operates on your most recent output document from List Processing, which is identified in the Current Output Document Block on the List Processing Menu.

NOTE: Sort Processing has no effect on the Current Output Document Block.

To print an output document after running List Processing:

1. Enter P (Print).
2. Make sure the information displayed on the print form is correct. If not, change the information.

3 Press RETURN to print

OR

If necessary, follow the instructions on your screen to change print settings.

Sort Processing

To use Sort Processing:

1 Create a list document and a sort specification document as discussed in Chapters 2 and 6.

2 Enter SR (Sort Processing) from the List Processing Menu.

These forms are displayed:

- Selecting List Document
- Selecting Sort Specification Document

NOTE: After you run Sort Processing once, the Sort Processing Documents Selected Form displays when you enter the Sort Processing option. Make sure the documents identified on the form are the ones you want to use. To add or change a document, press Gold A to display the document selection forms. When you change information on a form, use RUB WORD OUT if you need to erase a number that appears in the number field.

Completing the Document Selection Forms

Each document selection form has a field for folder name, document title, and document number.

To complete a form:

1 Enter the folder name and title of the document you want to process.

Use the TAB and HYPH PUSH keys to move from field to field.

2 Press RETURN to display the next form.

When you have completed these forms, the Select Output Document Option Menu displays. The options on this menu are described in Table 7-2. If you enter O, B, or T, the Selecting Output Document Form displays. If you enter N, the Creating New Output Document Form displays.

To complete either form:

1 Type the folder name and title of the output document.

   Use the TAB and HYPH PUSH keys to move from field to field.

2 Press RETURN.

   The Sort Processing Documents Selected Form displays.

How to Start Sort Processing

To start Sort Processing:

Press RETURN at the Sort Processing Documents Selected Form.

NOTE: You cannot use your terminal for other tasks while Sort Processing is running.

While you run Sort Processing, you see a running tally of:

- The number of records processed
- The number of records reproduced

In addition, the screen displays:

- The number of fields used in sorting
- The number of records in the list document

You can use this information to locate errors. For example, if you are sorting by two fields, but the Sort Processing Documents Selected Form shows 1 as the number of fields specified for sorting, the sort specification document may contain an error.

When Sort Processing is completed, WPS-PLUS displays the List Processing Menu.

To stop Sort Processing at any time:

Press CTRL/C.

Glossary

accumulator field

A field that keeps a running total.

comments

Optional text included in a *list document* that is not part of any record. Comments usually explain the content of the list document.

customized format

The type of format produced when you use a *format string*.

default format

The format List Processing Math uses for results if you do not use a *format string*.

end record marker (<>)

The left and right angle bracket characters, <>, with no text between them used to mark the end of a record in a *list document*.

field

The basic unit of information in a record. A field consists of a *field name* and a *field value*. For example: <FIELD 1>10.

field name

A name enclosed in angle brackets, <>, and followed by a *field value* in a *list document*. Used in a *form document* to tell List Processing where to insert field values.

field value

The text or numeric value that follows a field name in a *list document record*.

form document

The document that tells *List Processing* how to format information from a *list document*. The form document can contain text, such as a form letter, to merge with list document information.

format string

A string of characters enclosed in quotation marks at the end of a FORMULA command that tells *List Processing Math* how to format results.

input field

A field that contains numbers used in *List Processing Math* calculations. Input fields are on the left side of the equals sign in FORMULA commands.

leader

The text preceding the <!S> command at the top of a form document. The leader appears at the top of the *output document* after *List Processing*.

list document

A document containing a list of *records*. In *List Processing*, selected records from the list document are merged with a *form document* to produce *output documents* such as form letters.

List Processing

The WPS-PLUS feature that formats information from a *list document* as specified by a *form document* to produce copies of an *output document*. One use of List Processing is to produce personalized form letters.

List Processing Documents Selected Form

The form that WPS-PLUS displays during *List Processing*. The List Processing Documents Selected Form displays the current selected:

- List document
- Form document
- Specification document
- Output document

List Processing Math

The WPS-PLUS feature that performs calculations on data you include in a *list document*.

List Processing Menu

The menu that displays *List Processing* and *Sort Processing* options.

list specification document

A document that tells *List Processing* which *records* to process in a *list document*.

missing field

A *field name* common to other *records* that does not occur in a specific record in a *list document*.

numeric field name

A field name that starts with a left angle bracket and colon, such as <:COST>.

output document

The document that receives the output from *List Processing*. The output document contains text and formatting instructions for many printed reports, letters, envelopes, or labels.

record

A group of related *fields* in a *list document*, such as:

```
<name>John Doe
<addr>P.O. Box 1
<city>Merrimack
<state>NH
<zip>03054
< >
```

In this example, the five *fields* contain information about a person. *Field names* are in angle brackets, < and >, followed by *field values*. An end record marker, < >, marks the end of a record.

result field

A field *List Processing Math* uses for the result of a calculation. A result field is on the right side of the equals sign in FORMULA commands.

Selecting Form Document Form

A form that lets you select a *form document* for *List Processing*.

Selecting List Document Form

A form that lets you select a *list document* for *List Processing*.

Selecting List Specification Document Form

A form that lets you select a *list specification document* for *List Processing*.

Selecting Output Document Form

A form that lets you select an *output document* for *List* or *Sort Processing*.

Select Output Document Option Menu

A *menu* that displays the *List Processing* and *Sort Processing* output options.

Selecting Sort Specification Document Form

A form that lets you select a *sort specification document* for *Sort Processing*.

sort output document

The document that receives the output from *Sort Processing*. The sorted list document contains sorted list document records. You can use the sort output document as a list document when you run *List Processing*.

Sort Processing

The WPS-PLUS feature that arranges the records in a *list document* in ascending or descending order. The sorting is based on the contents of one or more of the *fields* in each record.

Sort Processing Documents Selected Form

The WPS-PLUS form that displays during *Sort Processing*. The Sort Processing Documents Selected Form displays the current selected:

- List document
- Sort specification document
- Output document

sort specification document

A document that tells Sort Processing how to sort the records in a *list document* during *Sort Processing*.

trailer

The text following the <!E> command in a *form document*. The trailer appears printed at the bottom of the *output document* after *List Processing*.

Index

<>command, 3-9, 3-16

A

Abbreviation document, 2-6
Angle brackets
  for defining records, 1-4
  for testing digits, 4-9
  printing, 3-20

C

Characters
  multinational, 1-5, 6-10
  technical, 1-5, 6-10
Columns, 3-16
Command,
  see name of each command
Comments, 2-5
Continuous-form
  envelopes, 3-5
  labels, 3-7
  letterhead, 3-3
Conventions, xiii
Creating lists
  see List document
Creating New Output Document Form, 7-6
CTRL/C, 7-9
Customized format, 5-15, GL-1

D

Default format, 5-14, GL-1
Document
  see also Form document, List document, List specification document, Output document, Sort specification document
  abbreviation, 2-6
  form, 1-2, 3-1, GL-2
  library, 2-6
  list, 1-2, 2-1, GL-3
  list specification, 1-2, 4-1, GL-4
  output, 1-2, GL-4

sort output, 1-11, GL-6
sort specification, 1-10, 6-1, GL-6

E

<!E> command, 3-9
End record marker (<>), 1-4
Envelopes, 3-5
Errors
List Processing, 7-9
List Processing Math, 5-19
Sort Processing, 7-11

F

Field names, 2-2, GL-2
assigning, 2-2
changing, 3-19
characteristics of, 2-2
Field values, 2-3, GL-2
missing, 2-5
Fields, 2-2, GL-2
accumulator, 5-9, GL-1
input, 5-5
missing, 2-5
numeric, 4-8, 5-5, GL-4
result, 5-5
Finish setting, 7-8
Form document, 1-2, 3-1, GL-2
columns in, 3-16
creating, 1-5
formatting guidelines, 3-2
leader, 3-9, GL-3
summary of commands, 3-9
trailer, 3-9, GL-7
types of, 3-2
Form document commands
<> command, 3-9, 3-16
<!E> command, 3-9
<!R> command, 3-9, 3-18
<!S> command, 3-9
Form letters, 3-3
Format
customized, 5-15, GL-1
default, 5-14, GL-1
Formatting
see also Form document
envelopes, 3-5
form letters, 3-3
labels, 3-7
leaders, 3-9
letterhead, 3-3
mailing lists, 3-8
tables, 3-8
tables, by grouping entries, 3-15
tables, with columns, 3-16
tables, with double spacing, 3-13
tables, with single spacing, 3-13
trailers, 3-9
FORMULA command, 1-7, 5-3
checklist, 5-20
guidelines, 5-5

G

Gold A, 7-3
Gold L, 7-4

H

Help, 1-7
HYPH PUSH, 7-4

L

Labels, 3-7
Leaders, 3-9, GL-3
Letterhead, 3-3
Library document, 2-6
List document, 1-2, 2-1, GL-3
changing field names in, 3-19

COMMENT command, 2-5
comments in, 2-5, GL-1
copying, 3-18
creating, 1-4
end record marker (<>), 1-4, GL-2
field names, 2-2, GL-2
field values, 2-3, GL-2
guidelines, 2-3
missing field values, 2-5
missing fields, 2-5, GL-4
moving records in, 1-5
numeric fields in, 2-4
order of fields in, 2-4
order of records in, 2-4
size, 2-4
skipping records, 3-18
List Processing, 1-1, GL-3
see also List document, Form document, List specification document, Output document
errors, 7-9
running, 7-1
starting, 7-7
List Processing Documents
Selected Form, 7-7, GL-3
finish setting, 7-8
number of records in list document, 7-8
number of records processed, 7-8
number of records selected, 7-8
running tally, 7-8
start setting, 7-8
List Processing Math, 1-7, 5-1, GL-4
$, 5-16
,, 5-16
., 5-15
9, 5-15
accumulator field, 5-9, GL-1
averages, 5-12
characters in formulas, 5-6
commands, 5-2
compared with Editor Math, 5-2
control blocks, 5-3
customized format, 5-15, GL-1
decimal places, 5-15
default format, 5-14, GL-1
errors, 5-19
*, for even columns, 5-17
(), for negative numbers, 5-17
format checklist, 5-21
format string, 5-14, GL-2
FORMULA command, 1-7, 5-3
FORMULA Control Block, 1-8, 5-3
input fields, 5-5, GL-3
null control block, 5-4
numeric fields, 5-5
records selected for processing, 5-19
result fields, 5-5, GL-5
ROUND command, 5-3, 5-18
SET command, 5-3
starting, 5-3
stopping, 5-4
subtotals, 5-9
totals, 5-9
-, trailing, 5-16
TRUNCATE command, 5-3, 5-18
WPSMATH command, 5-3
List Processing Math commands
FORMULA command, 5-3
FORMULA command checklist, 5-20
FORMULA command guidelines, 5-5
ROUND command, 5-3
SET command, 5-3
TRUNCATE command, 5-3
WPSMATH command, 5-3

List Processing Math format characters
$, 5-16
,, 5-16
., 5-15
9, 5-15
*, for even columns, 5-17
(), for negative numbers, 5-17
-, trailing, 5-16
List Processing Menu, 7-1, 7-2, GL-4
Document Processing (DOC) option, 1-6, 7-2
Print Output Document (P) option, 1-7, 7-2, 7-9
Sort Processing (SR) option 1-12, 7-2, 7-10
Test Specification Document (T) option, 4-13, 7-2
List specification document, 1-2, 4-1, GL-4
AND after BUT NOT IF, 4-5
AND-condition, 4-3
angle brackets (<>), 4-9
BUT NOT IF condition, 4-4
creating, 1-6
errors, 4-13, 7-9
guidelines, 4-12
IF-statement, 4-2
missing field names, 4-6
OR, 4-5
OR LESS, 4-10
OR MORE, 4-10
OR-condition, 4-3
PROCESS RECORD, 1-6
rejecting records by field value, 4-4
restricting records processed, 4-2
selecting a record by more than one value, 4-3
successive BUT NOT IFs, 4-5
testing character-for-character, 4-9
testing dates, 4-11
testing digits only, 4-9
testing for a specific number of characters, 4-7
testing for any single character, 4-6
testing for empty fields, 4-5
testing for errors in, 4-13
testing for partial values, 4-6
testing nonnumeric fields, 4-11
testing numeric values, 4-8
testing with numeric ranges, 4-10
THROUGH, 4-10
<*> wild card, 4-6
<?> wild card, 4-6
Lists, creating
see List document

M

Mailing list, 3-8
Multinational characters, 1-5, 6-10

O

Output document, 1-2, GL-4
double spacing, 3-13
grouping output from several records, 3-15
leader, 3-9
printing, 7-9
single spacing, 3-13
trailer, 3-9

P

Printing
  envelopes, labels, and nonstandard size letterhead, 3-2
  output documents, 7-9
Process up to and including, 7-8
Processing lists
  see List Processing

R

<!R> command, 3-9, 3-18
Records, 2-2, GL-5
ROUND command, 5-3
RUB WORD OUT, 7-3, 7-10

S

<!S> command, 3-9
Select Output Document Option Menu, 7-4, 7-11, GL-6
  B option, 7-5
  N option, 7-5
  O option, 7-5
  T option, 7-5
Selecting Form Document Form, 7-3, GL-5
Selecting List Document Form, 7-3, GL-5
Selecting List Specification Document Form, 7-3, GL-5
Selecting Output Document Form, 7-6, GL-5
Selecting records for List Processing
  see also List specification document
  finish setting, 7-8
  start setting, 7-8
Selecting Sort Specification Document Form, 7-10, GL-6
SET command, 5-3
Sort output document, 1-11, GL-6
Sort Processing, 1-10, 7-10, GL-6
  see also List document, Sort specification document, Sort output document
  errors, 7-11
  missing field values, 6-10
  missing fields, 6-10
  multifield, 6-2
  Sort Alphabet, 6-9
  sort in ASCENDING order, 6-6
  sort in DESCENDING order, 6-8
  starting, 7-10
Sort Processing Documents Selected Form, 7-10, GL-6
  fields used in sorting, 7-11
  number of records in list document, 7-11
  running tally, 7-11
Sort specification document, 1-10, 6-1, GL-6
  AND, 6-2
  creating, 1-12
  IN ASCENDING ORDER, 1-12
  IN DESCENDING ORDER, 1-12
  SORT ON, 1-12
  words and phrases used in, 6-10
Sorting lists
  see Sort Processing
Start processing from this record, 7-8
Start setting, 7-8

T

TAB, 7-4
Tables, 3-8
  spacing entries in, 3-13

Technical characters, 1-5, 6-10
Testing
see also List specification document
character-for-character, 4-9
dates, 4-11
digits only, 4-9
for a specific number of characters, 4-7
for any single character, 4-6
for empty fields, 4-5
for partial values, 4-6
list specification documents, 4-13
nonnumeric fields, 4-11
numeric values, 4-8
with numeric ranges, 4-10
Trailers, 3-9, GL-7
TRUNCATE command, 5-3

W

Word and Document Processing Menu
Create (C) option, 1-4
List Processing (LP) option, 1-6, 7-1
WPSMATH command, 5-3

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA and Puerto Rico call **800–258–1710**

In Canada call **800–267–6146**

In New Hampshire, Alaska or Hawaii call **603–884–6660**

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: P&SG Business Manager

INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION
P&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

*Any prepaid order from Puerto Rico must be placed with the Local Digital Subsidiary:
809–754–7575

Reader's Comments

Your comments and suggestions are welcome. They will help us in our continuous effort to improve the quality and usefulness of our publications.

1. How would you rate this manual for:

	good	fair	poor
completeness of information	_____	_____	_____
accuracy of information	_____	_____	_____
ease of use (clarity, organization)	_____	_____	_____

2. Did you find errors in this manual? Please specify by page and paragraph, or mark corrections on pages and attach copies to this form.

	page	paragraph
a) Incorrect information:	_____	_____
b) Information left out:	_____	_____
c) Hard to understand:	_____	_____

3. What suggestions do you have for improving this manual?

_____

_____

_____

4. What parts of the manual were especially good at helping you understand things?

_____

_____

_____

5. Please check the boxes that apply to you.

- ☐ Technical
- ☐ Nontechnical
- ☐ Management
- ☐ Nonmanagement
- ☐ Other (explain) _____
- ☐ Used other word processors
- ☐ Used computers
- ☐ Use the word processing unit yourself

Name _____ Title _____

Company _____

Street _____ City _____

State _____ Zip _____ Date _____

Do Not Tear - Fold Here and Tape

digital

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN: DISG Documentation ZKO2–2/N53
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, N.H. 03062

Do Not Tear - Fold Here and Tape